uc3m | Universidad Carlos III de Madrid

Grado Universitario en Ingeniería Informática
2017/2018

Trabajo Fin de Grado

Análisis, diseño e implantación de solución de apoyo a la toma de decisión basado en el análisis de publicaciones en redes sociales.

Jesús Guillén Encinas

Tutor

Alejandro Calderón Mateos

6 Marzo 2018

ABSTRACT

INTRODUCTION.

The purpose of this work is to use a BI tool that uses the information generated by twitter, thus making a dashboard for the decision making of the different positions of responsibility of a company. This tool will be used by communication managers and community managers.
A few decades ago the data was still manageable, currently those responsible for making business decisions, marketing, and business cannot deal directly with the data, but they need a tool that displays and shows in a concise and summarized only the relevant data. This is where Business Intelligence is important; it is a set of tools and procedures capable of transforming many data into information. It facilitates the analysis to generate competitive impact in the market.

Twitter has sophisticated ways of communicating; making less than 150 characters contain all the data the communicator wants to transmit, making a valuable source of information. For companies whose impact on social networks is very high, tools are necessary to give meaning to the massive amount of tweets generated by an advertising campaign, news related to the brand or any event that generates a reaction from users and public opinion.

This work has two objectives: first and foremost, to build an environment that transforms data into useful information in a graphic way. For this, different parameters and variables provided by twitter API will be analyzed and those that are valuable will be chosen. The second objective is the implementation of a system that can cover the data of an advertising campaign of one or more brands to study the total impact and the real need to have a dashboard to manage the data.

STATE OF THE ART.

The BI tools are responsible for giving information for decision making in companies. The decision making is based on accurate and timely information and guarantees to choose the most convenient alternative for the success of the company.

Currently most companies and organizations have staff in charge of marketing and advertising. These support large amounts of data, being able to extract most of the data from social networks. Through the information generated after the processing of the data they can argue future decisions.

Following Twitter statistics [5].

- User statistics: There are currently 310 million active users per month, having created in 2016 a total of 1.3 billion accounts. The average number of followers of a user is 208.
- Usage statistics: More than 500 million tweets per day are sent, equivalent to 6000 tweets / second.
- Marketing statistics: 65.8% of companies in the US with more than 100 employees use the platform to do marketing. It is also estimated that 77% of the users who are answered by the community managers have a positive feeling when they are answered. 58% of the brands have more than 100,000 followers and 80% of users have mentioned a brand in a tweet.

SOFTWARE ANALYSIS.

For the realization of this project a system is needed that provides in a contrasted way information extracted from the data. In the following table it can be seen the comparison of exposed solutions for later implementation.

	No tools.	**ELK**	**Power BI**
Tool Price.	-	Free Software.	0 euros to 4200 node/month
Learning difficulties.	-	High	Low
Data extraction.	No	Yes	Yes
Documentation.	-	Developing.	High availability.
Deployment.	-	Hard.	Easy.

Tabla 1: Software.

Confirming the use of a tool for the reliability of the information when talking about large amounts of data, there is the use of Microsoft Power Bi against the stack of tools formed by Elasticsearch, Logstash and Kibana.

ELK is shown in the project, for meeting the requirements of the project. It is also a compact tool that fits perfectly with the type of problem addressed.

SOFTWARE DESIGN.

A tool or stack of tools is necessary to meet the following requirements:

1. Collect data.
2. Text search engine to filter the information.
3. Visualization of data in graphic form.

For the requirement number 1 there are two solutions that can cover the problem.

	Platform	**Event Routing**	**Plugin Ecosystem**	**Transport**	**Performance**
Logstash	Linux & Windows	Algorithmic statements	Centralized	Deploy with Redis for reliability.	Uses more memory. Use Elastic Beats for leafs.
Fluentd	Linux & Windows	Tags	Decentralized	Built-in reliability but hard to configure.	Uses less memory. Use Fluent Bit and Fluentd Forwarder for leafs.

Ilustración 1: Logstash & Fluentd.

As it can be seen both can be used in both Linux and Windows, as well as being free software.

	Solr	**Elasticsearch**
Installation and Configuration	Supported by detailed documentation	More intuitive
Indexing/Searching	Text-oriented	Better performance of analytical queries
Scalability and Clustering	Provides SolrCloud	Better inherent scalability and designed for the cloud
Community	A much bigger ecosystem of community	A growing community though not a complete open source mindset
Documentation	Very well-documented	Lacks in documentation

Ilustración 2: Solr & Elasticsearch.

These two search engines are the most used for non-relational indexing. Also, because they meet all the requirements, one of the two must be selected. In this case, being Elasticsearch part of a complete system with Logstash and having a more intuitive configuration and using is taken advantage of by the use of this. In addition both have a strong system of cloud computing.

For requirement number 3, Grafana and Kibana are compared, the two most important systems for visualizing data. Although Kibana belongs to the elastic stack Grafana should not be discarded due to its high performance.

In this case it can be therefore chosen both solutions, because they are very similar tools, and if there is a mastery of both it can be chosen one or another solution.

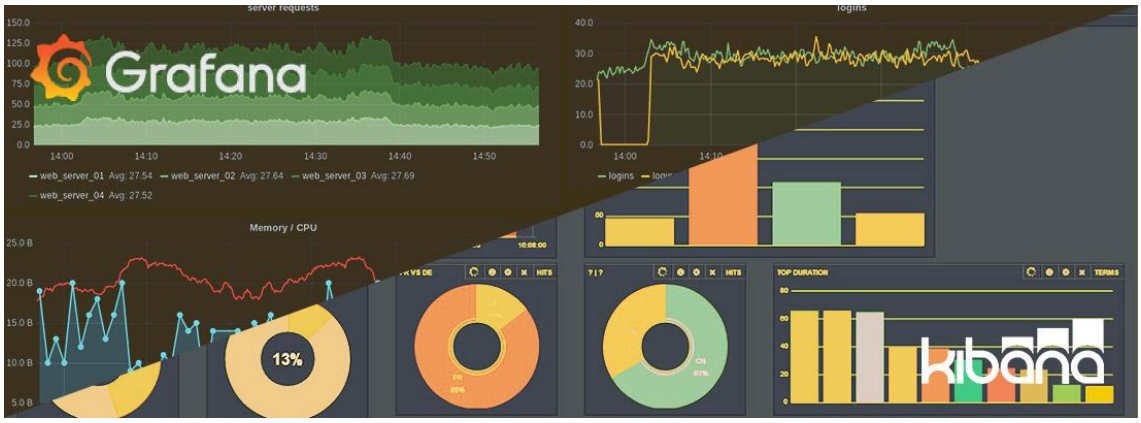

Ilustración 3: Grafana & Kibana.

In this project, Kibana is used, due to its strong connection with Elasticsearch.

Logstash is an open source data processing tool. It ingests a large amount of data, transforms it and provides it to a search engine. In this case Elasticsearch.

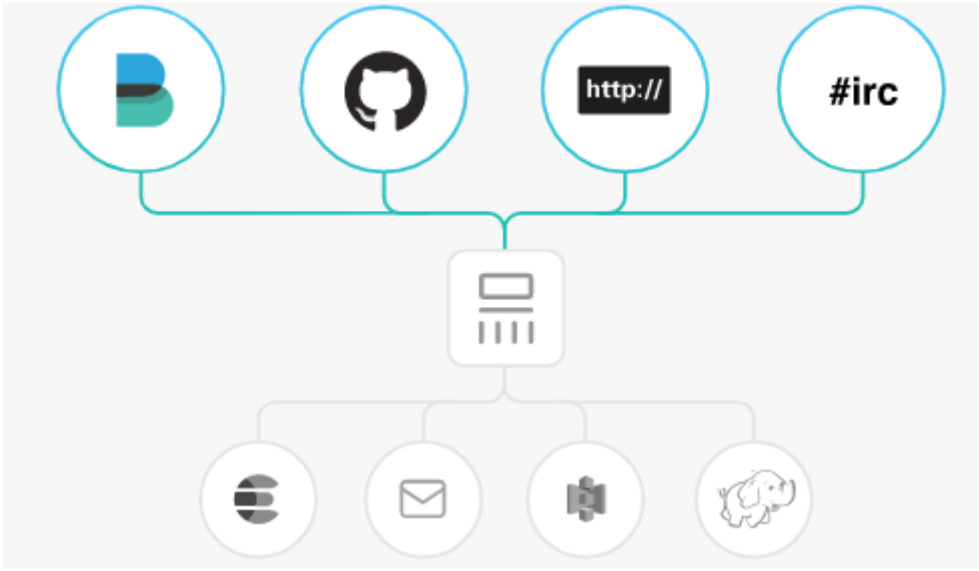

Ilustración 4: ELK.

The data is normally scattered, so Logstash allows a wide variety of entries, including Twitter. In addition, thanks to the filters provided, it is capable of performing tasks such as facilitating processing independently of the source, format or scheme.

Elasticsearch is the main pillar on which the ELK stack is based, it is a text search engine, with a RESTful web interface and JSON documents. It is able to combine all types of searches at the same time allowing to explore trends and patterns in the data.

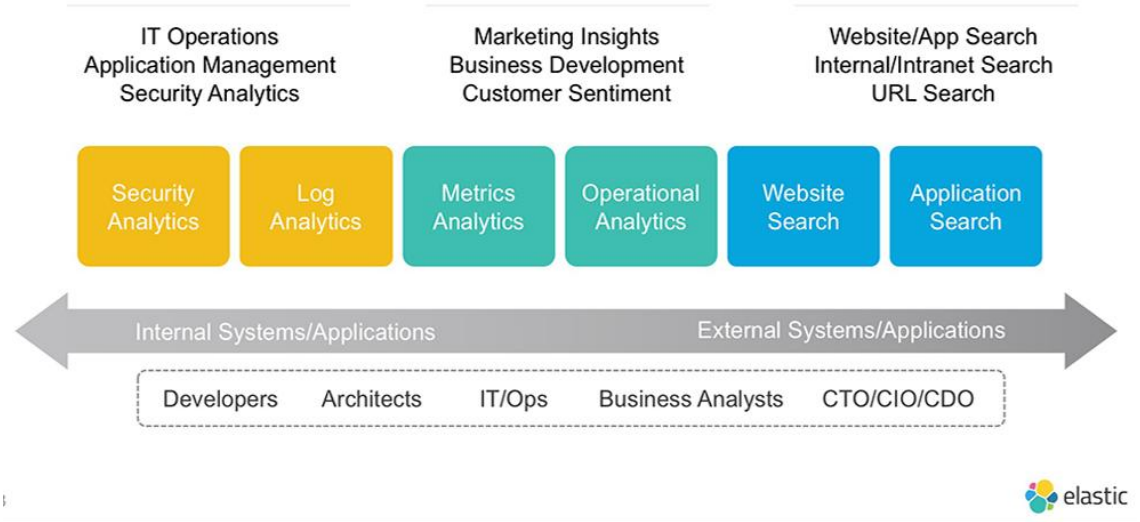

Ilustración 5: Applications

Kibana allows viewing Elasticsearch data. It is a Dashboard creation tool that looks like this.

Ilustración 6: Dashboard.

It has chosen to use the Elastic cloud solution. The architecture is similar and the important points of the design will be detailed below.

Ilustración 7: Architecture.

The illustration shows the basic structure to follow.

The necessary data is extracted from Twitter. Logstash is responsible for this process, which is defined in the input through its own structure. The body of this tool is defined by input-filter-output. As a filter, the use of JSON will be explained in the implementation, so it parses the information obtained and it sends through output to elasticsearch. Note that Twitter is a service in the cloud, like elasticsearch and kibana

(until now you had been capitalized), while logstash is being executed on a local physical machine.

Elasticsearch, under an AWS environment, is in charge of indexing the information that Logstash sends in real time and this, in turn, is in charge of supplying the information indexed to kibana when consulting any metric to graph.

An illustration of the cloud system is shown below.

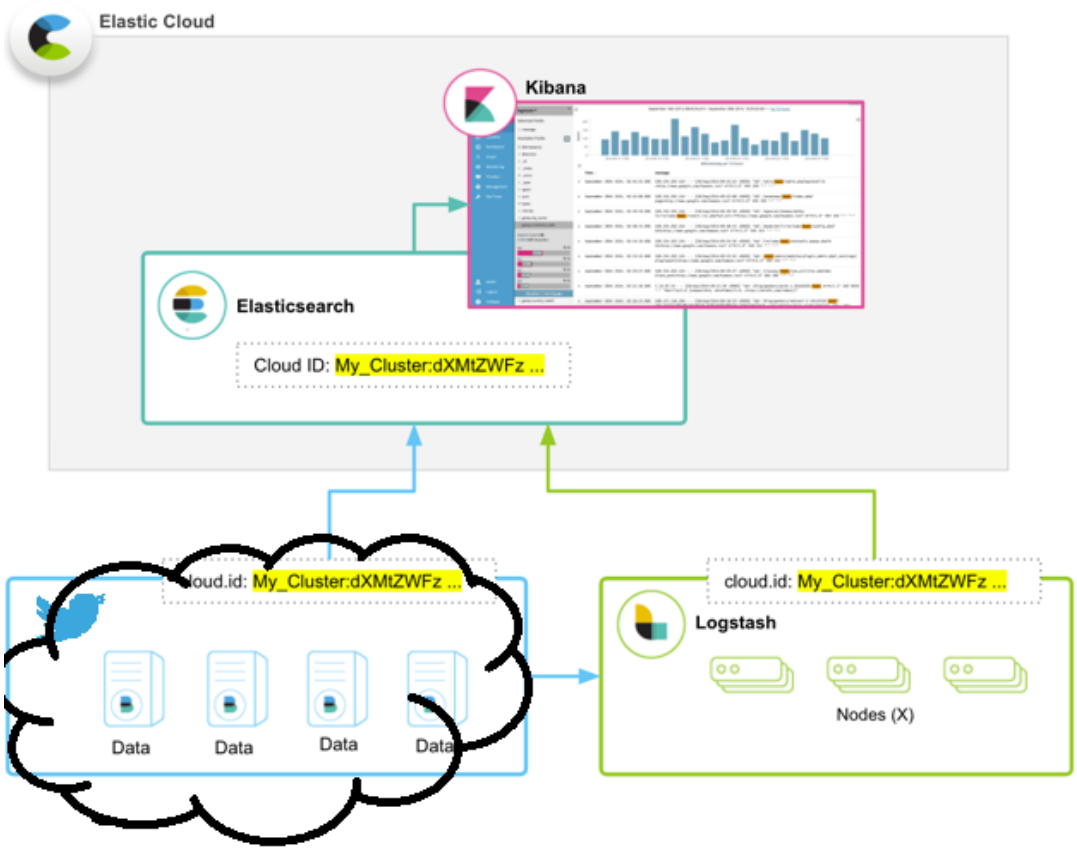

Ilustración 8: Elastic Cloud.

Due to the nature of the ELK stack the approach is divided into five sub-sections:

1. Data collection.
2. Structuring and filtering information.
3. Information display.
4. Update of the information.
5. Extraction of knowledge.

The achievement of all the sub-sections except for the update (real-time data) makes the application fully functional.

SOFTWARE INSTALLATION

The solution works independently of the operating system and the browser.

In this particular case, the following tools have been selected:

- The operating system on which Logstash is installed is Debian 9.3.x The version of Logstash, Elasticsearch and Kibana is 5.6 (last version on January 10, 2018 stable).
- Putty version 0.70 has been used to connect to the machine that contains Logstash.
- For the storage of Elasticsearch + Kibana, AWS platform for ELK has been used.
- For the Kibana interface, Google Chrome 64.0.x browser has been used.

PLANNING

At this point the planning applied during the development of the project will be detailed making use of two Gantt diagrams, one with planning and the other with real time.

In addition a budget for the project will be established in a real way, without trial periods and discarding the academic field.

At this point the chosen model is detailed, in this case a cascade model with feedback in which in each of the steps you can advance or go back to the starting point.

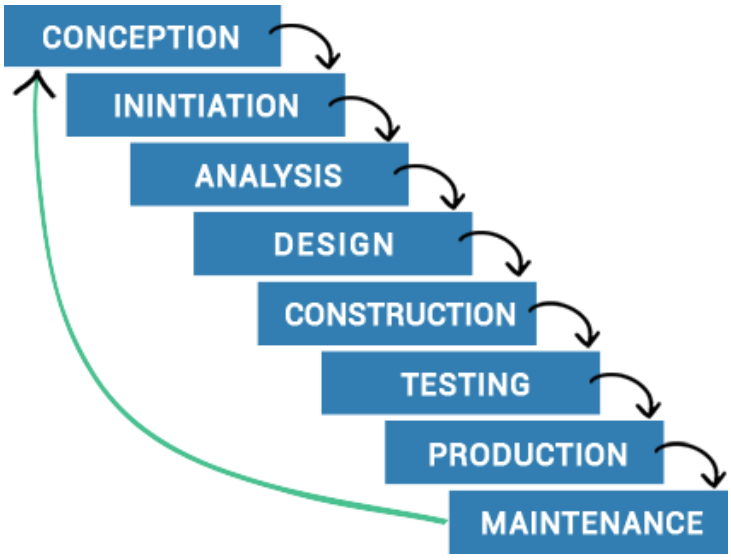

Ilustración 9: Cascade.

This allows after finding faults in each of the stages can be scaled back to make changes in the previous stages.

The initial term for the realization of this project was estimated on July 1, 2017 to be completed in a maximum of 8 months, date of presentation of the project.

For this, a system was carried out that divided the processes in estimation of difficulty.

The initial Gantt is detailed below.

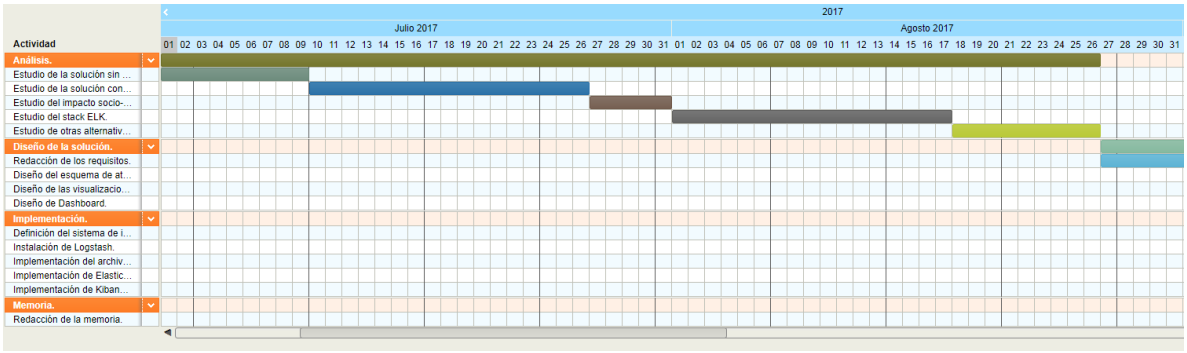

Ilustración 10: Gantt inicial 1.

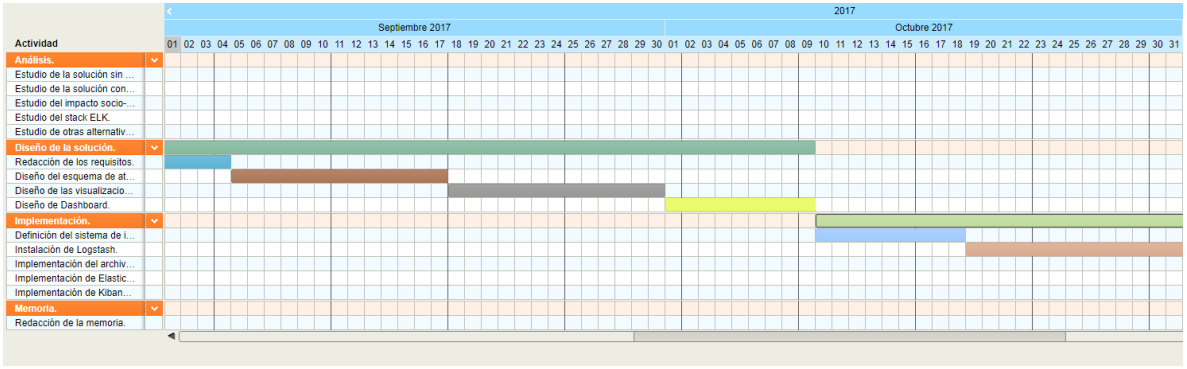

Ilustración 11: Gantt inicial 2.

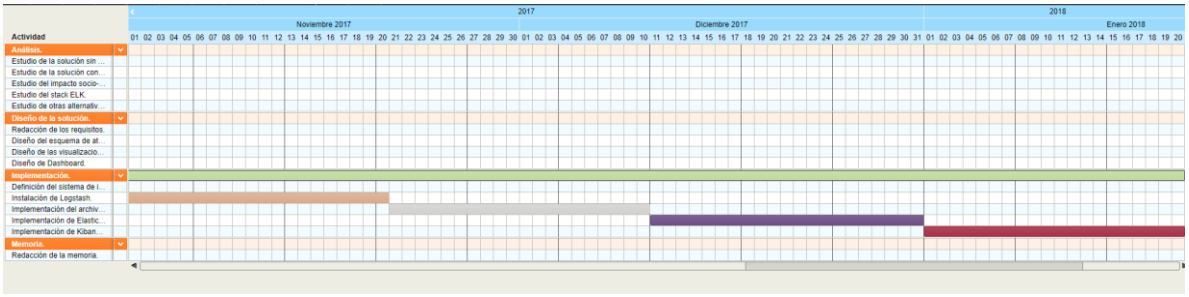

Ilustración 12: Gantt inicial 3.

After the development of the project, the real time differs in some points, so that both the time table and the Gantt chart are restructured, giving the following result.

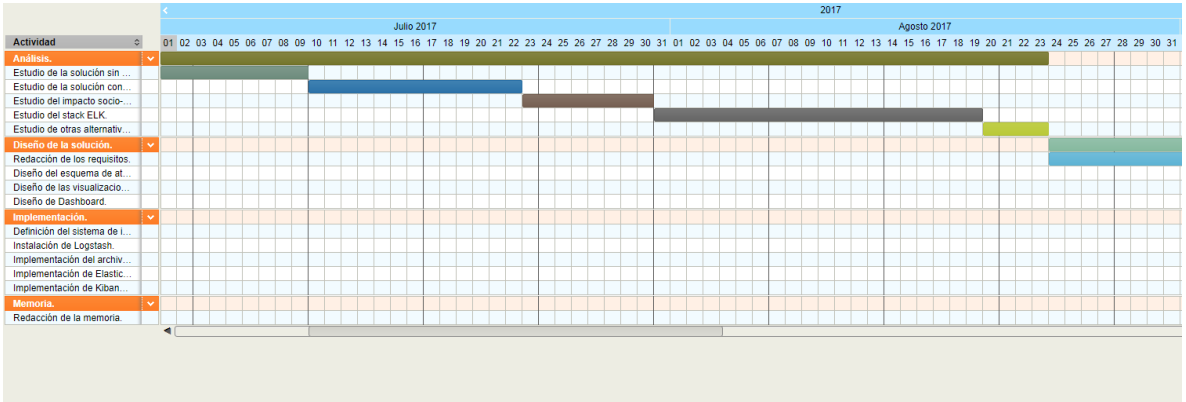

Ilustración 13: Gantt final 1.

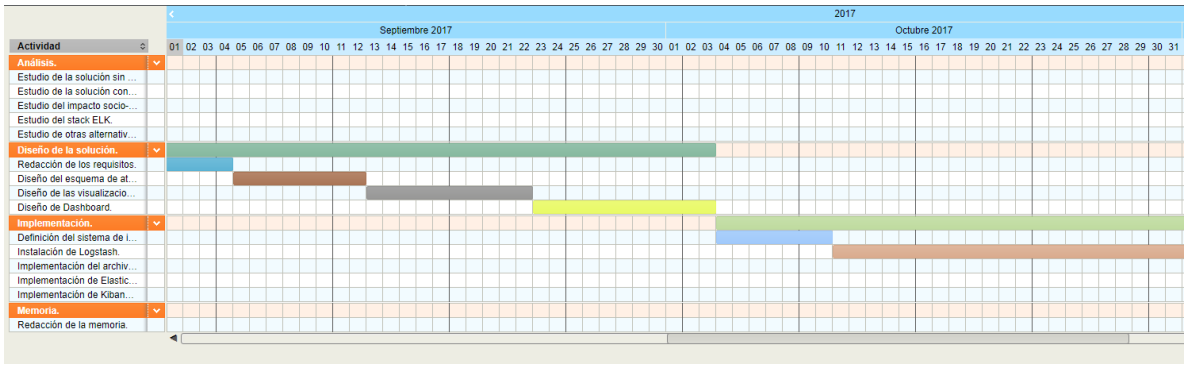

Ilustración 14: Gantt final 2.

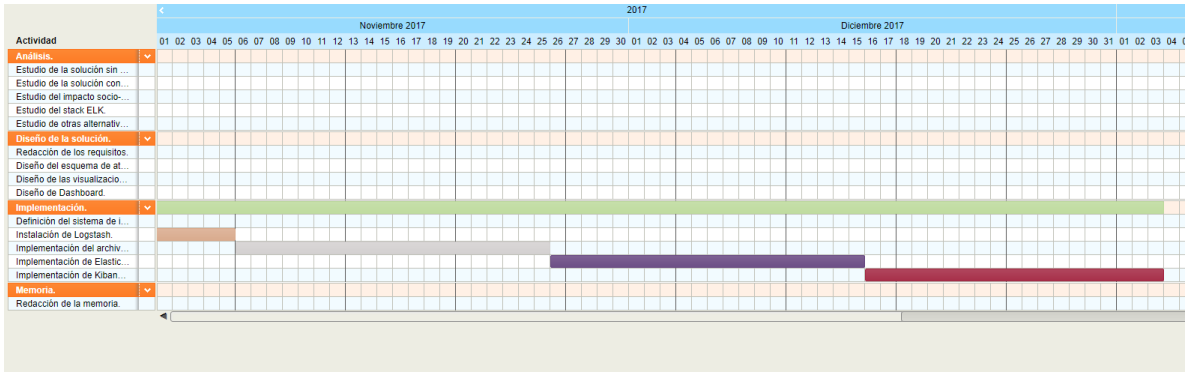

Ilustración 15: Gantt final 3.

CONCLUSIONS

The work revolves around Elasticsearch and the stack of tools that encompass it, so we can observe a tendency to use text search engines in non-relational systems, whose data intake is quite large.

The ELK system is not easy to understand at the beginning since each tool can be used unilaterally, while being part of a whole. So it can be difficult to understand how to perform the job of fitting the data.

However, when the process is understood and the logs are observed, investigating the errors can even be corrected and be part of the community that creates the system, providing new solutions.

ELK has been chosen because it is undoubtedly the system that comes closest to the initial approach, in which it was necessary to extract the data from Twitter and transform it (Logstash), then store it in a structured way (Elasticsearch) and finally show a dashboard (Kibana) .

At first was thought of developing the idea for Logs generated by computers of the Carlos III University of Madrid. After this initial approach, it was gone through the analysis process. Multiple articles have been written about ELK and other BI tools for log processing had been written, so this work fell short. The idea of the need of a person was born to compare itself in RRSS with the others, in terms of numbers of followers, reactions before the publications, etc.

This finally evolved to the point of reflection on who are the most interested in knowing a certain type of information with massive amounts of data, so they opted to use it in advertising systems on Twitter, that is, tweets about a particular brand.

After this process, a Twitter account and a Twitter application were created to collect the necessary Tokens.

Once established the connection with Twitter and linked with Elasticsearch, it is a matter of practice and reading of manual to be able to create any type of visualization that you want with the obtained data.

Personal Conclusions.

Personally, this project has been the biggest one I've been involved in, making it possible to experience first-hand from minute zero how to install completely unknown software for me until I can provide solutions to the ELK community today.

I would definitely go back to choose this project, perhaps focusing it from other points of view, but always under ELK and the same support from colleagues and tutor.

FUTURE WORK

There are multiple options in terms of future work, with the same idea of the project, the requirements would change, making important other ideas not obtained here, for example the time when an account is more named.

Besides that it is not a unique system it can be combined with others like Beats or Grafana.

In this aspect Beats could provide the tweets to Elasticsearch and Grafana to paint under this scheme.

A future work that had no place for the difficulty and time of the project is the incorporation of feelings analysis of tweets, there are tools that are responsible for given a text analyze the feeling that describes, so Logstash could be used to extract tweets, providing it to one of these tools as a plugin in the filter definition of its body and giving a new parameter depending on the mood.

DEDICATORIA

A Papá.
A Mamá.
A Isa.

A los que llevan desde que empezamos esta travesía.
A los que se fueron por voluntad propia (hicisteis de mi alguien mucho más fuerte).
A Alba, por hacer de ella un lugar donde ser feliz.
A Ana, por el magnífico absurdo de los domingos.
A Raquel, por soñar juntos y al mismo tiempo.
A Javi, por darle sentido a la palabra amistad.
A Marta, que se merece más que nadie que le agradezca su apoyo incondicional.

A Alejandro Calderón, por su ejemplo, porque profesores como él, dan sentido a esta carrera de fondo.

¡¡GRACIAS!!

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

Palabras clave: ELK, Redes sociales, toma de decisiones, Business Intelligence, tuit, Twitter, Community manager, KPI, marca, cuenta de Twitter, Dashboard.

1 Introducción.

1.1 Motivación.

En las últimas décadas se ve una tendencia al cambio en el ámbito de la toma de decisiones (previamente tomadas únicamente por un experto o grupo de expertos en la materia). Esto es debido a la forma exponencial que toman los datos.

Mientras que un par de décadas atrás los datos todavía eran *"manejables"*, actualmente los encargados de elegir una vía (ya sea comercial, de marketing, empresarial, etc.) no pueden tratar directamente con los datos, sino que necesitan de una herramienta que visualice y muestre de forma concisa y resumida únicamente los datos relevantes. Es aquí donde la **Inteligencia de Negocios** (BI - *Business Intelligence*) toma forma y sentido, dando un conjunto de herramientas y procedimientos capaces de transformar la ingente cantidad de datos en información. Facilita por tanto el análisis para generar impacto competitivo en el mercado.

Twitter ha sofisticado la forma de comunicarse, haciendo que en poco menos de 150 caracteres se contengan todos los datos que desea transmitir el comunicador. Haciendo de este contenedor una fuente valiosa de información. Para empresas cuyo impacto en **redes sociales** es muy alto, son necesarias herramientas que den sentido a la cantidad masiva de tuits que genera una campaña de publicidad, una noticia relacionada con la marca o cualquier evento que genere una reacción de los usuarios y la opinión pública.

Bajo la premisa de los párrafos anteriores, nace el sentido de este trabajo. Utilizar una **herramienta de BI** que se nutra de la información que genera *Twitter*, haciendo así un cuadro de mandos (*Dashboard*) para la toma de decisiones de los distintos cargos de responsabilidad de una empresa. Esta herramienta estará destinada tanto a los responsables de comunicación como a los *"community manager"*.

1.2 Objetivos.

Este trabajo contempla dos objetivos: el primero y principal, construir un entorno que transforme datos en información útil de una forma gráfica. Para esto se analizarán los diferentes parámetros y variables que brinda la API de *Twitter* y elegir aquellos que sean valiosos. Como segundo objetivo se tiene la implementación de un sistema que pueda abarcar los datos de una campaña publicitaria de una o varias marcas para ver el impacto total y la necesidad real de disponer de un cuadro de mandos para manejar la ingesta de datos.

1.3 Alcance.

Este proyecto abarca varias tareas:

- **Análisis** de campañas publicitarias y alcance de las mismas.
- **Estudio** de las diferentes herramientas de Business Intelligence.
- Análisis y elaboración de la arquitectura de la herramienta seleccionada.

1.4 Destinatarios.

Este proyecto se encuentra dirigido a aquellos responsables de marketing y/o publicidad de empresas medianas y grandes que deseen una solución para el análisis, transformación y procesado de la información, dándoles la oportunidad de modelar los datos, haciendo así de estos algo mucho más valioso.

1.5 Estructura del documento.

Este documento se estructura de la siguiente forma:

- **Introducción**: Apartado en el que se incluye una breve descripción del trabajo desarrollado.
- **Estado del Arte**: Apartado en el que se explica cómo se encuentra el panorama actual de las tecnologías utilizadas desde su inicio. Además de incluir de forma más detallada el *stack* de herramientas utilizadas.
- **Análisis**: Apartado en el que incluimos los requisitos de sistema.
- **Diseño**: Apartado dedicado al diseño de la arquitectura del sistema y a la toma de decisiones sobre este.
- **Implementación e implantación**: Apartado en el que se desarrolla la implementación del *stack* de *elastic cloud*.
- **Planificación y presupuesto**: Apartado dedicado a la planificación y tiempo de desarrollo. Además de un presupuesto que abarca la elaboración del proyecto en su totalidad.
- **Conclusiones**: Apartado con las principales conclusiones extraídas del trabajo.
- **Bibliografía**: Apartado contenedor de las reseñas bibliográficas.

2 Estado del arte.

La Inteligencia de Negocios BI (*Business Intelligence*) [1] es una herramienta bajo la cual diferentes tipos de organizaciones pueden soportar la toma de decisiones basadas en información precisa y oportuna, garantizando la generación del conocimiento

necesario que permita escoger la alternativa que sea más conveniente para el éxito de la empresa.

En la actualidad la mayoría de las empresas y organizaciones en general cuentan con personal a cargo del marketing y publicidad de su marca, estas además soportan grandes cantidades de datos, pudiendo extraer mucha de estos de **redes sociales**. Mediante la información generada tras el procesamiento de los datos pueden argumentar las decisiones futuras.

2.1 Impacto socio-económico.

Se puede observar el impacto socio-económico en [3]: "Un estudio realizado en Europa por *Information Builders Ibéric* mostró el costo que tiene la falta de sistemas de toma de decisiones en las organizaciones, según estos datos, el empleado europeo medio pierde una media de 67 minutos diariamente buscando información de la compañía, lo que equivale a un 15,9% de su jornada laboral. Para una organización de 1.000 empleados que gane unos 50.000 euros al día esto equivale a 7,95 millones de euros al año de salario perdido, todo ello por la búsqueda de información para tomar una decisión."

Es trivial entonces extrapolar los datos de empresas de 1.000 empleados a empresas de marcas conocidas multinacionales que llegan al orden de 100.000 empleados, donde el ahorro podría llegar en torno a 100 millones de euros al año, pudiendo invertir en los departamentos de I+D+i que normalmente se relacionan con el trabajo de marketing e informática.

Queda expuesto por tanto el poder competitivo que da la implementación de sistemas basados en BI.

2.2 Conceptos de BI.

2.2.1 Inteligencia de negocio.

La inteligencia de negocio se define como la habilidad que tienen las empresas de tomar decisiones. Mediante una correlación de datos, estos pueden ser extraídos y transformados de forma que den sentido único generando información. De este modo se tendrá conocimiento sobre los problemas u oportunidades a las que se enfrente la compañía, mejorando y aprovechando al máximo la toma de decisiones.

2.2.2 Dashboard.

El *dashboard* o cuadro de mandos es una representación gráfica de la información que resulta interesante y que está orientado a la toma de decisiones para realizar una estrategia sólida. Finalmente lo que se quiere es transformar datos en información y esta a su vez en conocimiento.

2.2.3 Información actual de Twitter.

Siguiendo las estadísticas que proporciona [5]:

- Estadísticas de usuario: Existen actualmente 310 millones de usuarios activos al mes, habiéndose creado en 2016 un total de 1,3 mil millones de cuentas de las cuales escriben al menos un tuit antes de su cierre el 44%. El promedio de seguidores de un usuario es de 208.

- Estadísticas de uso: Son mandados más de 500 millones de tuits al día, un equivalente a 6000 tuits/segundo.

- Estadísticas de marketing: El 65,8% de las empresas en EEUU con más de 100 empleados utilizan la plataforma para hacer marketing. Además se estima que el 77% de los usuarios que son contestados por los *community manage*r tienen un sentimiento positivo cuando son contestados. El 58% de las marcas tienen más de 100000 seguidores. Un 80% de los usuarios han mencionado alguna marca en un tuit.

2.3 Trabajos similares.

Actualmente hay numerosos trabajos relacionados con sistemas de inteligencia de negocios, tanto para la toma de decisiones en empresas como para particulares, ya sea con carácter financiero, educativo, etc.

En el ámbito de toma de decisiones utilizando como medio las redes sociales se encuentra "Redes Sociales Visuales: Caracterización, Componentes y posibilidades para el SEO de Sitios Intensivos en Contenidos" [6] que muestra como segundo objetivo del proyecto vincular las redes sociales visuales como *Instagram* con el *SEO*.

3 Análisis.

Para la realización de este proyecto se necesita un sistema que proporcione de forma contrastada información extraída de los datos. En la siguiente tabla se puede observar la comparativa de soluciones expuestas para su posterior implementación.

	Sin herramientas	ELK	Power BI
Precio de la herramienta.	-	Software Libre	Desde 0 euros en su versión más económica hasta 4200 por nodo/mes
Dificultad de aprendizaje.	-	Alta	Baja
Extracción fiable en una cantidad de datos exponencial.	No	Sí	Sí
Documentación.	-	En desarrollo.	Muy buena.
Despliegue.	Ninguno.	Costoso.	Fácil.

Tabla 2: Herramientas.

Descartando el no uso de una herramienta por la fiabilidad de la información cuando se habla de grandes cantidades de datos, queda la utilización de *Microsoft Power Bi* frente a la pila de herramientas formadas por *Elasticsearch, Logstash y Kibana*, que en el resto del documento *se nombrarán* como ELK.

Este último es el que se mostrará en el proyecto, por cumplir los requisitos que se detallan a continuación en el punto 3.1, además de ser una herramienta compacta que encaja a la perfección con el tipo de problema abordado.

3.1 Requisitos.

Para definir los requisitos se seguirá el estándar IEEE-STD-830-1998 que permite la definición mediante el uso de tablas.

Identificador: R(N)F-XX			
Nombre			
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción			

Tabla 3: Tabla de requisitos.

Los atributos que definen un requisito son:

- Identificador: código unívoco que identifica cada requisito.
- Nombre: descripción simple del requisito.
- Prioridad: medida que determina la urgencia del requisito.
- Necesidad: medida del interés en el requisito.
- Estabilidad: medida sobre el valor de permanencia del requisito.
- Descripción: Explicación del requisito.

3.1.1 Requisitos funcionales.

RF-01			
Nombre	Visualización de cuentas.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar el número de cuentas a las que llega un tuit.		

Tabla 4: RF-01

RF-02			
Nombre	Visualización de nombramientos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar el número de veces que una cuenta ha sido nombrada.		

Tabla 5: RF-02

RF-03			
Nombre	Visualización de comparativa por cuentas.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar la comparativa de marcas en cuanto a número de cuentas a las que llega un tuit.		

Tabla 6: RF-03

RF-04			
Nombre	Visualización de comparativa por nombramientos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar la comparativa de marcas en cuanto a número de veces que ha sido nombrada.		

Tabla 7: RF-04

RF-05			
Nombre	Visualización de lugares.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar el porcentaje de los lugares desde los que se nombra la marca.		

Tabla 8: RF-05

RF-06			
Nombre	Visualización de cuentas verificadas.		
Prioridad	Alta	**Media**	Baja
Necesidad	Alta	**Media**	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar el número de cuentas verificadas desde los que se nombra la marca.		

Tabla 9: RF-06

RF-07			
Nombre	Visualización de top países.		
Prioridad	Alta	Media	**Baja**
Necesidad	Alta	Media	**Baja**
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar los países que más han nombrado la marca en orden descendente.		

Tabla 10: RF-07

RF-08			
Nombre	Visualización de top palabras.		
Prioridad	Alta	Media	**Baja**
Necesidad	Alta	Media	**Baja**
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá visualizar las palabras más nombradas en relación a la marca.		

Tabla 11: RF-08

RF-09			
Nombre	Filtrado por visualización de cuentas.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá filtrar por RF-01 las cuentas que se deseen.		

Tabla 12: RF-09

RF-10			
Nombre	Filtrado por nombramientos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá filtrar por RF-02 los nombres que se deseen.		

Tabla 13: RF-10

RF-11			
Nombre	Filtrado por cuentas.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá filtrar por RF-03 y RF-04 las cuentas que se deseen.		

Tabla 14: RF-11

RF-12			
Nombre	Número de cuentas.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El número de cuentas que podrán ser filtradas en RF-11 es dos.		

Tabla 15: RF-12

RF-13			
Nombre	Filtrado por lugares.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá filtrar por RF-05 el porcentaje de un lugar específico.		

Tabla 16: RF-13

RF-14			
Nombre	Filtrado por cuentas verificadas.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá filtrar por RF-06 por cuentas verificadas.		

Tabla 17: RF-14

RF-15			
Nombre	Filtrado por top paises.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá filtrar por los países elegidos en RF-07.		

Tabla 18: RF-15

RF-16			
Nombre	Número de top paises.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El número de países de RF-07 será 5.		

Tabla 19: RF-16

RF-17			
Nombre	Número de filtrado por top paises.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El número de países por los que se podrá filtrar en RF-15 es uno.		

Tabla 20: RF-17

RF-18			
Nombre	Cambio de cuenta.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá cambiar de cuenta.		

Tabla 21: RF-18

RF-19			
Nombre	Cambio de cuenta.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá cambiar de cuenta.		

Tabla 22: RF-19

RF-20			
Nombre	Cambio de color de gráficos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá cambiar el color de los gráficos.		

Tabla 23: RF-20

RF-21			
Nombre	Cambio de título de gráficos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá cambiar el título de los gráficos.		

Tabla 24: RF-21

RF-22			
Nombre	Cambio de título de dashboard.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá cambiar el título del dashboard.		

Tabla 25: RF-22

RF-23			
Nombre	Cambio de tipo de gráficos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Se podrá cambiar el tipo de gráficos.		

Tabla 26: RF-23

3.1.2 Requisitos no funcionales.

RNF-01			
Nombre	Red social.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	La red social utilizada será Twitter.		

Tabla 27: RNF-01

RNF-02			
Nombre	Tiempo real.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El sistema dará información en tiempo real.		

Tabla 28: RNF-02

RNF-03			
Nombre	Software Libre.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El sistema estará basado en herramientas de Software Libre.		

Tabla 29: RNF-03

RNF-04			
Nombre	Web.		
Prioridad	Alta	Media	**Baja**
Necesidad	Alta	Media	**Baja**
Estabilidad	Alta	**Media**	Baja
Descripción	El sistema estará plataformado en web.		

Tabla 30: RNF-04

RNF-05			
Nombre	Color de los gráficos.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El sistema estará basado la psicología del color [6] tomando Tonos neutros de azul violeta y gris		

Tabla 31: RNF-05

RNF-06			
Nombre	Título corto.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El título de los gráficos no puede ser mayor de 20 caracteres.		

Tabla 32: RNF-06

RNF-07			
Nombre	Título descriptivo.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El título de los gráficos será descriptivo.		

Tabla 33: RNF-07

RNF-08			
Nombre	Tipo de gráficos Basic Charts.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El tipo de gráficos deberá estar incluido en la siguiente lista: • Area. • Heat Map. • Horizontal Bar. • Line. • Pie. • Vertical Bar.		

Tabla 34: RNF-08

RNF-09			
Nombre	Tipo de gráficos Data.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El tipo de gráficos deberá estar incluido en la siguiente lista: • Data Table. • Gauge. • Goal. • Metric.		

Tabla 35: RNF-09

RNF-10			
Nombre	Tipo de gráficos Maps.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El tipo de gráficos deberá estar incluido en la siguiente lista: • Coordinate Map. • Region Map.		

Tabla 36: RNF-10

RNF-11			
Nombre	Tipo de gráficos Time Series.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El tipo de gráficos deberá estar incluido en la siguiente lista: • Time Series.		

Tabla 37: RNF-11

RNF-12			
Nombre	Tipo de gráficos Markdown.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El tipo de gráficos deberá estar incluido en la siguiente lista: • Markdown.		

Tabla 38: RNF-12

RNF-13			
Nombre	Filtrado texto.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	Todas las filtraciones que se llevan a cabo en RF-9 RF-10 RF-11 RF-13 RF-14 RF-15 RF-16 RF-17 serán por texto.		

Tabla 39: RNF-13

RNF-14			
Nombre	Sistema operativo.		
Prioridad	**Alta**	Media	Baja
Necesidad	**Alta**	Media	Baja
Estabilidad	**Alta**	Media	Baja
Descripción	El sistema tendrá que ser soportado tanto por Windows como por Linux.		

Tabla 40: RNF-14

3.2 Aspectos legales.

Respecto al marco regulador no existe ningún imperativo legal que pueda aplicarse a este trabajo, ya que la información obtenida es totalmente pública. No obstante se recalcan dos puntos importantes.

Este trabajo se encuentra bajo una licencia *Creative Commons* Reconocimiento – No Comercial – Sin Obra Derivada.

Ha de tenerse cuidado con los datos obtenidos y qué tipo de información se obtiene, respetando siempre la información sensible, como puede ser la orientación sexual, ideología política, etc.

4 Diseño.

Tras definir los requisitos que debe tomar el sistema se procede al diseño de una solución.

4.1 Elección de las herramientas.

Como se describe en los apartados 2 y 3 es necesario un mecanismo o pila de herramientas que cumplan los siguientes requisitos:

4. **Recolecta de datos** vía *Twitter*.
5. **Motor de búsqueda de texto** para filtrar la información.
6. **Visualización de datos** de forma gráfica.

Para el requisito número 1 se disponen dos soluciones que pueden abarcar el problema.

	Platform	Event Routing	Plugin Ecosystem	Transport	Performance
Logstash	Linux & Windows	Algorithmic statements	Centralized	Deploy with Redis for reliability.	Uses more memory. Use Elastic Beats for leafs.
Fluentd	Linux & Windows	Tags	Decentralized	Built-in reliability but hard to configure.	Uses less memory. Use Fluent Bit and Fluentd Forwarder for leafs.

Ilustración 16: Logstash & Fluentd.

Como se puede observar ambas pueden ser utilizadas tanto en Linux como Windows, además de ser software libre.

Para el requisito número 2 se analizan las siguientes soluciones.

	Solr	Elasticsearch
Installation and Configuration	Supported by detailed documentation	More intuitive
Indexing/Searching	Text-oriented	Better performance of analytical queries
Scalability and Clustering	Provides SolrCloud	Better inherent scalability and designed for the cloud
Community	A much bigger ecosystem of community	A growing community though not a complete open source mindset
Documentation	Very well-documented	Lacks in documentation

Ilustración 17: Solr & Elasticsearch.

Estos dos motores de búsqueda son los más utilizados para la indexación no relacional, lo que hace que junto a que cumplen todos los requisitos se tenga que decantar por uno de los dos. En este caso, al ser *Elasticsearch* parte de un sistema completo junto a *Logstash* y tener una configuración y un uso más intuitivo se decanta por utilizar este.

Además ambos tienen un fuerte sistema en la nube que se detallará en el punto 4.1.2.

Para el requisito número 3 se comparan *Grafana* y *Kibana*, los dos sistemas más importantes para visualizar datos. Aunque *Kibana* pertenece al *stack* de *Elastic* no se debe descartar *Grafana* debido a su alto rendimiento.

En este caso se pueden elegir por lo tanto ambas. Se trata de herramientas muy parecidas, y si se tiene dominio de ambas se puede escoger una u otra solución por estética.

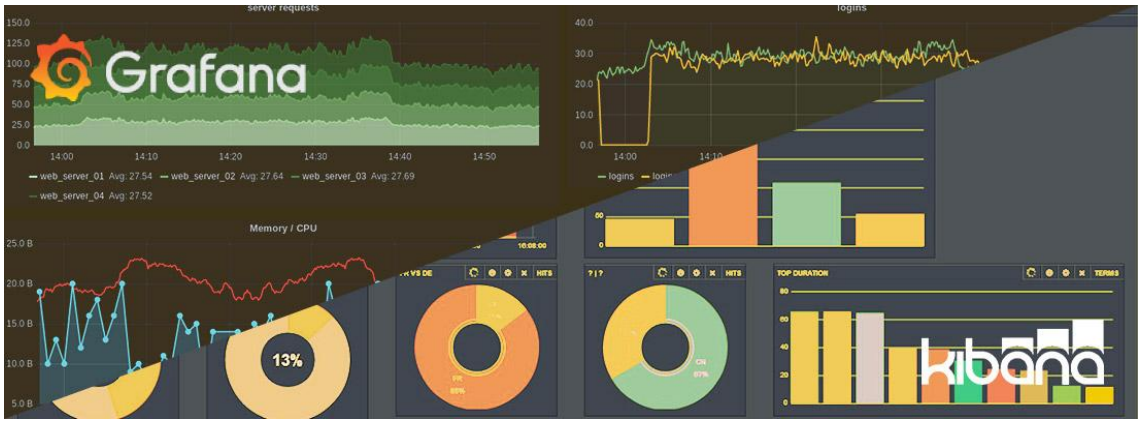

Ilustración 18: Grafana & Kibana.

En este proyecto se utiliza *Kibana* por lo detallado anteriormente y su fuerte conexión con *Elasticsearch*.

4.1.1 Logstash.

Logstash es una herramienta de procesamiento de datos de código abierto. Ingiere una gran cantidad de datos, los transforma y se los proporciona a un motor de búsqueda. En este caso a *Elasticsearch*.

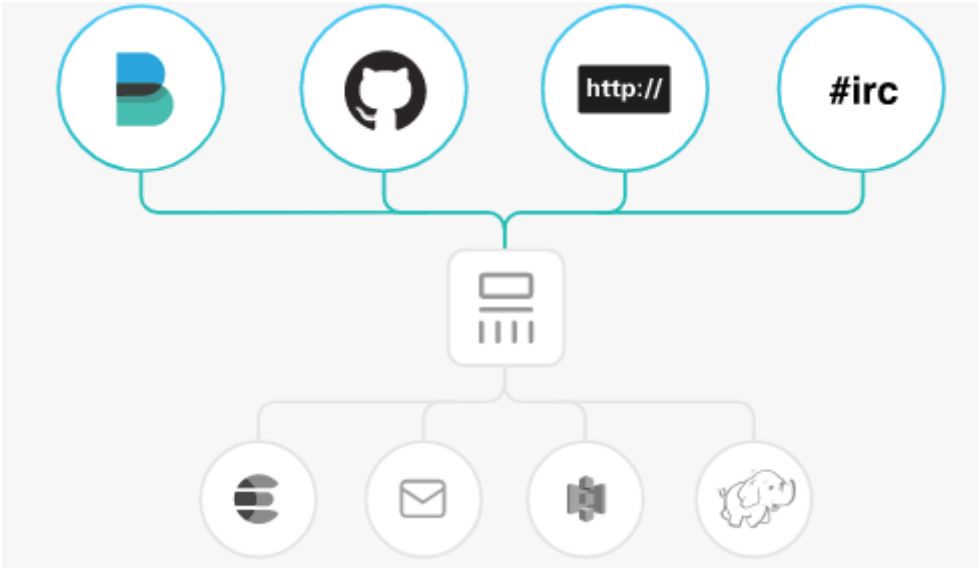

Ilustración 19: Data Elastic.

Los datos normalmente están dispersos, por lo que Logstash permite una gran variedad de entradas entre las que se encuentra Twitter.

Además gracias a los filtros proporcionados es capaz de realizar tareas como facilitar el procesamiento independientemente de la fuente, formato o esquema.

Ilustración 20: Esquema data.

Aunque se elija en este caso *Elasticsearch*, no es el único resultado al que se puede llegar, pudiendo transportar los datos a una variedad de resultados donde enrutar.

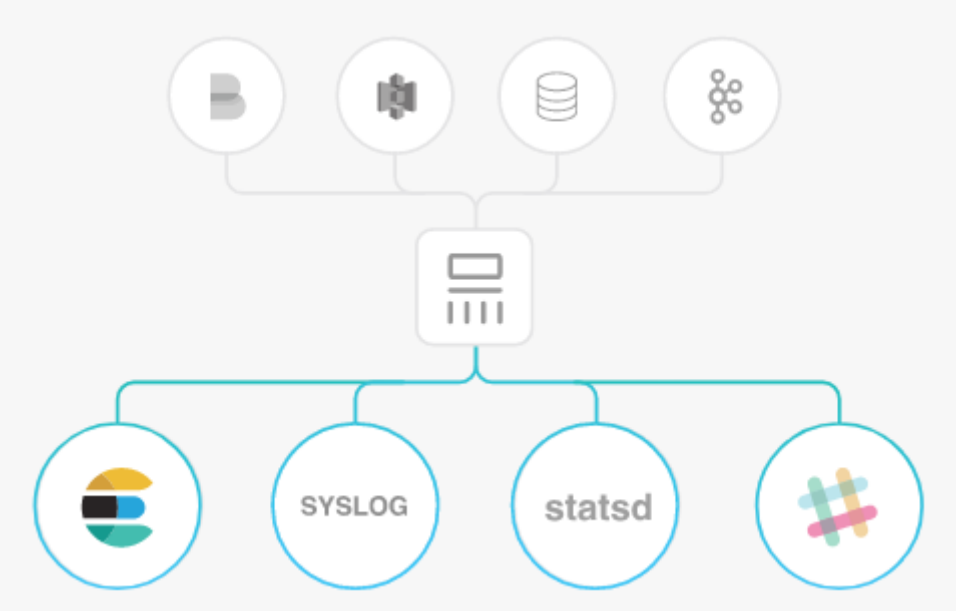

Ilustración 21: Sistemas de enrutamiento.

4.1.2 Elasticsearch.

Elasticsearch es el pilar principal en el que se apoya el *stack ELK*. Se trata de un motor de búsqueda de texto, con una interfaz web *RESTful* y documentos *JSON*, capaz de

combinar todo tipo de búsquedas a la par que permite explorar tendencias y patrones en los datos.

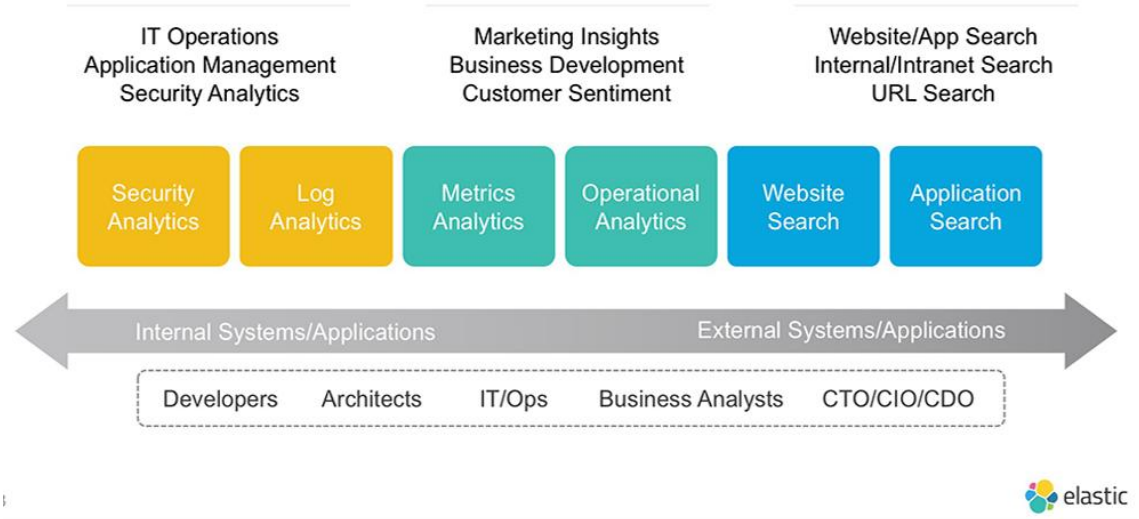

Ilustración 22: Elastic.

4.1.3 Kibana.

Kibana permite visualizar los datos de *Elasticsearch*. Es una herramienta de creación de *Dashboard* con el siguiente aspecto.

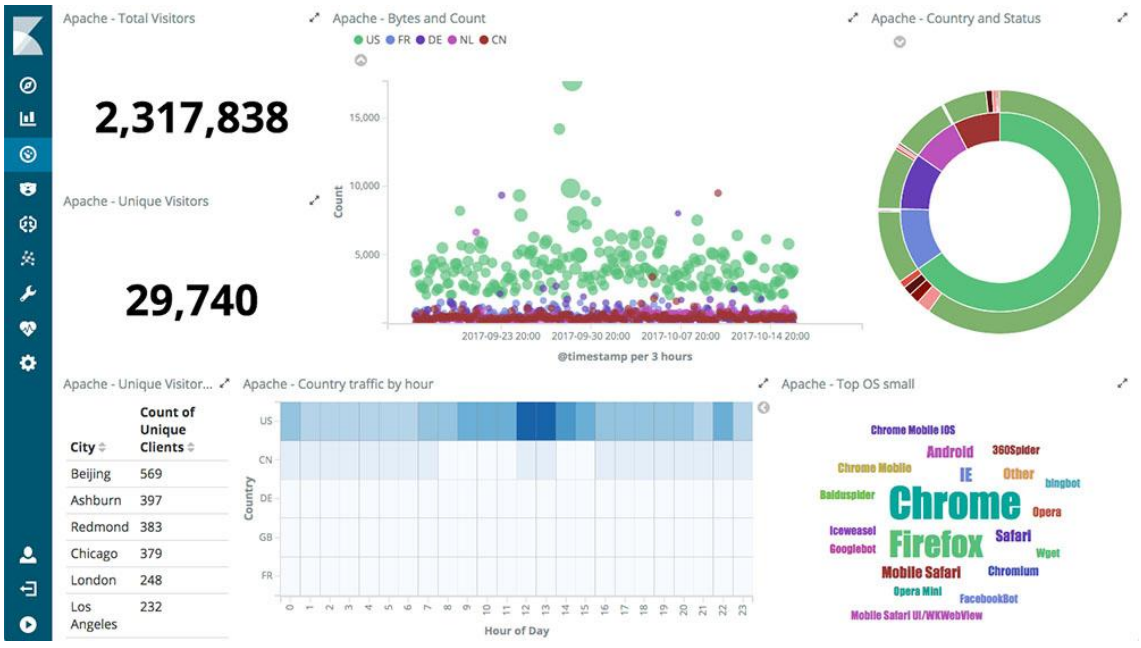

Ilustración 23: Kibana.

4.1.3 Coste de utilización.

Un sistema real tendría el siguiente coste para desarrollo en la nube bajo AWS.

Reserved memory	**32GB**
Reserved storage	768GB
High availability	Yes
Data centers	2
Price/hour	$2.3945
Price/month	$1748

Tabla 41: Coste de utilización.

Para el sistema desarrollado se han utilizado periodos gratuitos de 14 días renovados sistemáticamente copiando el *index* y *cluster* utilizados.

Reserved memory	**4GB**
Reserved storage	96GB
High availability	Yes
Data centers	2
Price/hour	$0.4029 – Free Trial 14 $0
Price/month	$250.75 – Free Trial 14 $0

Tabla 42: Coste Free Trail.

4.1.3 Facilidad de aprendizaje.

ELK proporciona una guía en la que se puede navegar interactivamente para consultar detalles que sean interesantes para el proyecto.

La calidad de la documentación es alta y se ofrecen tutoriales en la página web principal.

En concreto se aportan tres vídeos, no obstante para verlos es necesario aportar el email.

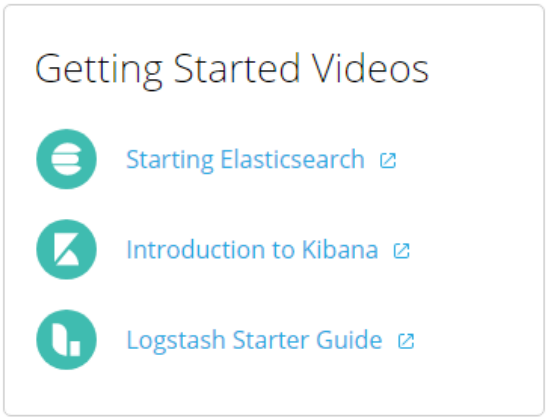

Ilustración 24: Tutoriales.

4.2 Diseño de la propuesta.

Se ha optado por utilizar la **solución *Elastic* basada en la nube**. La arquitectura es similar y se detallarán a continuación los puntos importantes del diseño.

4.2.1 Arquitectura y seguridad.

Ilustración 25: Arquitectura.

La ilustración mostrada muestra la estructura básica a seguir.

Se extraen los datos necesarios de Twitter. De este proceso se encarga *Logstash*, que mediante una estructura propia se define en el input. El cuerpo de esta herramienta viene definido por *input-filter-output*. Como filtro se explicará en la implementación el uso de JSON, por lo que parsea la información obtenida y se manda mediante *output* a *Elasticsearch*. Cabe destacar que Twitter es un **servicio en la nube**, al igual que *Elasticsearch* y *Kibana*, mientras que *Logstash* se está ejecutando en una máquina física en local.

Elasticsearch, bajo un **entorno *AWS***, se encarga de indexar la información que *Logstash* envía en tiempo real y este, a su vez, es el encargado de suministrar la información indexada a *Kibana* cuando consulte cualquier métrica para graficar.

A continuación se muestra una ilustración del **sistema *cloud***.

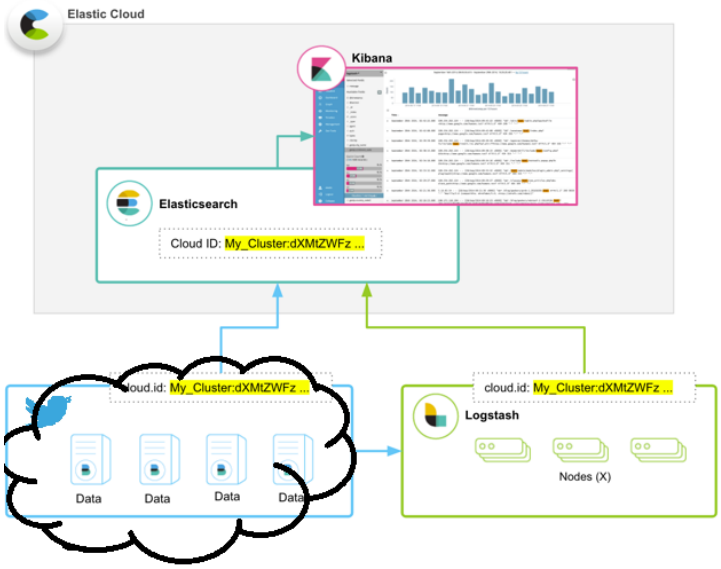

Tabla 43: Cloud.

En cuanto a seguridad, el sistema posee dos características básicas:

1. Protección de cluster para accesos no autorizados con contraseña y acceso basado en roles.

2. Integridad de los datos con autenticación de mensajes y *cifrado SSL / TSL*.

En la versión que se utiliza existe X-Pack con características propias de seguridad, alerta y monitorización.

Como se muestra en la siguiente ilustración, se pueden personalizar los roles, tanto a nivel de cluster como a nivel de índice.

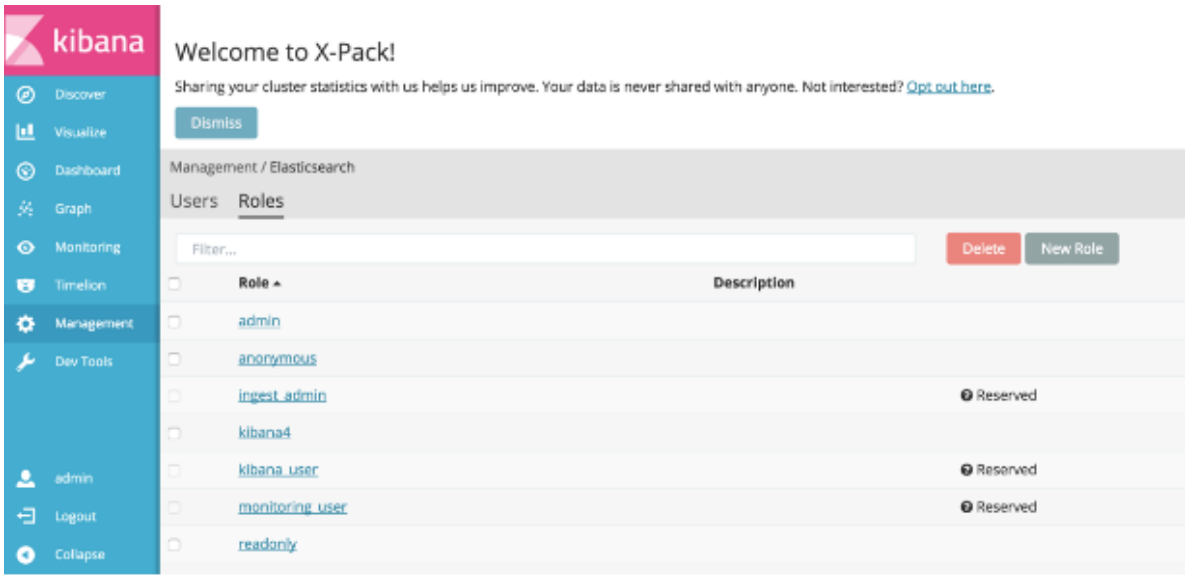

Ilustración 26: Roles Kibana.

4.2.1.1 Seguridad de la conexión Logstash - Twitter.

Para que el imput de Logstash pueda recolectar la información de una cuenta, esta previamente tiene que estar provista de una aplicación Twitter, la cual asigna los siguientes parámetros:

consumer_key => "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"

consumer_secret => "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

oauth_token => "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

oauth_token_secret => "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

Estas contraseñas están expuestas en texto plano en la ruta de la aplicación, por lo que si la máquina de instalación sobre la que se monta Logstash va a ser utilizada por otros usuarios que no sean root se debe proteger el archivo. En caso contrario con la protección de la máquina en sí es suficiente.

4.2.2 Planteamiento.

Debido a la naturaleza de la pila de *ELK* el planteamiento se divide en cinco subapartados:

1. Recolección de datos.
2. Estructuración y filtrado de la información.
3. Visualización de la información.

4. Actualización de la información.

5. Extracción de conocimiento.

La consecución de todos los subapartados a excepción de la actualización (datos en tiempo real) hace que la aplicación sea totalmente funcional.

4.2.2.1 Extracción de los datos.

El proceso de extracción de datos se realiza mediante *Logstash*. La *API* de *Twitter* se encarga de dar la estructura de la información. Dicha estructura es utilizada por *Elasticsearch* para generar el *index* que contendrá el *mapping* sobre el que posteriormente se trabajará con *Kibana*. Este *mapping* se compone de 1269 *fields* o documentos en el que cada uno de ellos comparte un sentido único.

Este proceso además funciona de tal manera que *Logstash* hace de tubería siempre abierta (es un sistema de **procesamiento en tiempo real**) por lo que *Elasticsearch* únicamente tiene que ir indexando de manera que *Kibana* dispone de las métricas instantáneamente.

4.2.2.2 Estructuración y filtrado de la información.

La información se estructura siguiendo el siguiente esqueleto:

```
PUT my_index (1)

{"mappings": {"doc": { (2)

                        "properties": { (3)

                                "title":   { "type": "text"  }, (4)

                                "name":    { "type": "text"  }, (4)

                                "age":     { "type": "integer" },  (4)

                        "created":  {

                                "type":   "date", (4)

                                "format": "strict_date_optional_time||epoch_millis"

 }       }       }       }
```

1. Creación del índice, en el caso de ejemplo *my_index*.

2. Añadido del *mapping* (propia estructura de *my_index*), en el caso de ejemplo.

3. Especificación y propiedades del documento (en el caso de ejemplo *propierties* o *fields*).

4. Especificación de cada tipo de dato para cada documento (*type*).

En cuanto a la filtración, *Twitter* proporciona su propio esqueleto por lo que la estructura de filtro de *Logstash* está vacía.

4.2.2.3 Visualización de la información.

Kibana ofrece los siguientes tipos de visualizaciones:

1. **Basic Charts.**

   - **Area**: Compara parámetros en coordenadas X/Y.

Ilustración 27: Area.

   - **Heat Map**: Mapea sombras dentro de una matriz.

Ilustración 28: Heat Map.

   - **Horizontal Bar**: Compara parámetros en coordenadas X/Y.

Ilustración 29: Horizontal Bar.

   - **Line:** Compara parámetros en coordenadas X/Y.

Ilustración 30: Line.

- **Pie:** Muestra la contribución de cada fuente a un total.

Ilustración 31: Pie.

- **Vertical Bar**: Compara parámetros en coordenadas X/Y.

Ilustración 32: Vertical bar.

2. **Data**.

- **Data Table:** Muestra los datos en bruto de una agregación.

Ilustración 33: Data table.

- **Gauge**: Muestra un medidor.

Ilustración 34: Gauge.

- **Goal**: Muestra un medidor.

Ilustración 35: Goal.

- **Metric**: Muestra un único número.

Ilustración 36: Metric.

3. **Maps**.

    - **Coordinate Map**: Asocia los resultados de una agregación con ubicaciones geográficas.

Ilustración 37: Coordinate map.

    - **Region Map**: Mapas en los que la intensidad de un color corresponde al valor de una métrica.

Ilustración 38: Region map.

4. **Time Series.**

    - **Timelion**: Calcula y combina datos de múltiples conjuntos de series temporales.

Ilustración 39: Timelion.

    - **Visual Builder**: Visualiza datos de series de tiempo usando agregaciones de canalizaciones.

Ilustración 40: Visual builder.

5. **Markdown**: Muestra información o instrucciones de forma libre.

Ilustración 41: Markdown.

6. **Tag Cloud**: Muestra palabras en una nube en la que el tamaño de cada palabra corresponde a su importancia (normalmente cuando más repetida está una palabra, mayor es su importancia).

Ilustración 42: Tag cloud.

4.2.2.4 Actualización de la información.

En este apartado se detalla la utilización de la pila *ELK* como herramienta de desarrollo, ya que es necesario un sistema en tiempo real, por lo que la actualización de la información es esencial y además debe de ser lo menos analógica posible.

Kibana ofrece un botón de actualización instantánea.

Ilustración 43: Auto-refresh.

Además se puede programar el rango de tiempo de actualización de tres formas diferentes: Una inicial llamada **Quick** donde seleccionamos rangos usualmente utilizados y ya definidos.

Ilustración 44: Quick.

Relative.

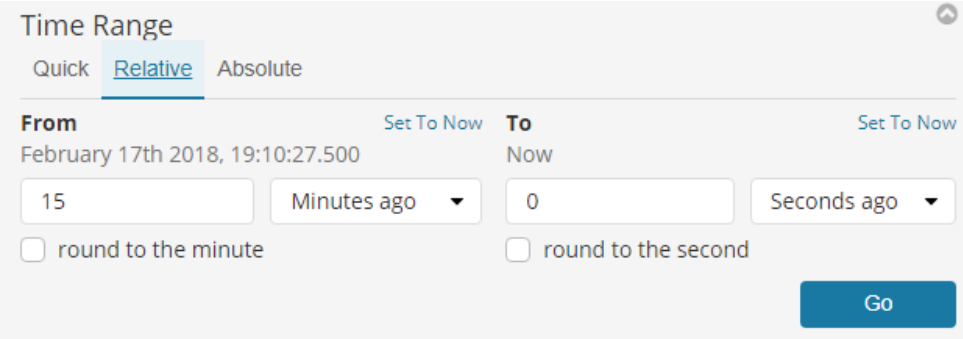

Ilustración 45: Relative.

Absolute.

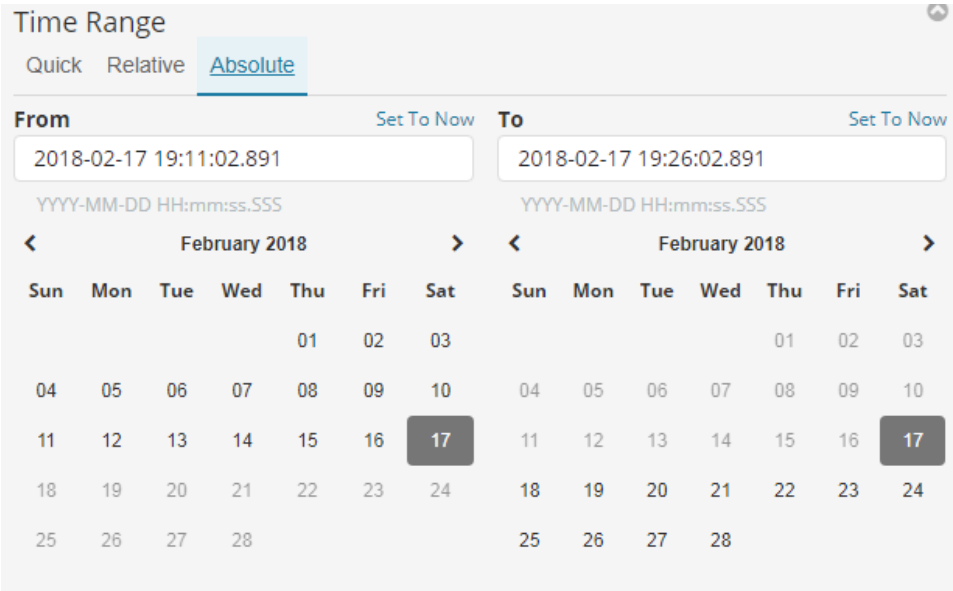

Ilustración 46: Absolute.

4.2.2.5 Extracción de conocimiento.

Para entender estos datos hace falta la colaboración de un experto, se puede tener la información y que esta carezca de sentido.

A efectos académicos no hará falta en este proyecto. No obstante, es aconsejable que aquel que defina las métricas y los indicadores sea el experto en publicidad, mientras que la persona encargada de ejecutar sea conocedor de la herramienta y sus funcionalidades.

Un ejemplo simple para entender esto sería la siguiente ilustración:

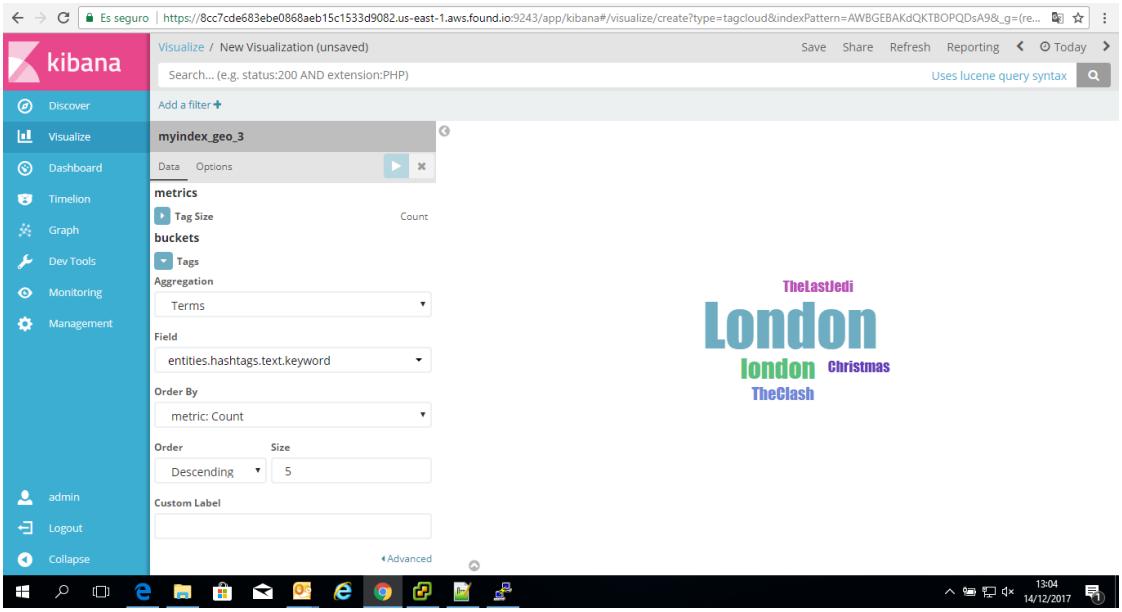

Ilustración 47: Consulta London.

Esta consulta está realizada el 14 de diciembre de 2017. En ella se tiene un *Tag Cloud* donde se filtra por *Londres*, *London* y *London*(palabras con el mismo significado). Debido a la fecha es normal que se repita la palabra christmas. En cuanto a *TheLastJedi*, y *TheClash* falta información para extraer el conocimiento y poder confirmar porqué esas dos palabras son repetidas. En el primer caso la instancia está ejecutada un día antes del estreno de la película *The Last Jedi*. Mientras que en el segundo caso se tiene que indagar un poco más para esclarecerlo, hace referencia a una banda británica de punk que tituló la canción *'London Calling'* este mismo día, por lo que si se desconoce el aniversario, esta información carecería de sentido (falta de conocimiento).

4.3 Casos de uso.

Finalizado el planteamiento del sistema, a continuación se detallan los casos de uso de la solución que definirán las actividades de los agentes externos.

Se proceden a especificar en forma de tabla siguiendo la siguiente estructura:

Identificador: CU-XX	
Nombre	
Actores	
Objetivo	
Precondiciones	
Postcondiciones	
Flujo normal	

Los atributos que definen un caso de uso son los siguientes:

- **Identificador**: Código unívoco para identificar el caso de uso.
- **Nombre**: Descripción simple.
- **Actores**: Actores que interactúan en el caso de uso.
- **Objetivo**: Finalidad del caso de uso.
- **Precondiciones**: Estado del sistema antes de realizar las acciones.
- **Postcondiciones**: Estado del sistema después de realizar las acciones.
- **Flujo normal**: Secuencia de acciones que se deben llevar a cabo.

CU-01	
Nombre	Visualización de cuentas.
Actores	Usuario.
Objetivo	Visualizar las cuentas a las que llega un tuit.
Precondiciones	Haber lanzado el tuit.
Postcondiciones	Se visualizan las cuentas a las que ha llegado el tuit.
Flujo normal	1. Iniciar sesión en la cuenta que se quiera utilizar. 2. Lanzar tuit. 3. Esperar el tiempo que se desee. 4. Refrescar el Dashboard o esperar al autorefresh.

Tabla 44: CU-01

CU-02	
Nombre	Visualización de nombramientos.
Actores	Usuario.
Objetivo	Visualizar el número de veces que una cuenta ha sido nombrada.
Precondiciones	
Postcondiciones	Se visualiza el número de veces que se ha nombrado una cuenta.
Flujo normal	1. Marcar el inicio de tiempo. 2. Marcar el final de tiempo. 3. Refrescar el Dashboard o esperar al autorefresh.

Tabla 45: CU-02

CU-03	
Nombre	Visualización de comparativa por cuentas.
Actores	Usuario de la cuenta 1. Usuario de la cuenta 2.
Objetivo	Visualizar las cuentas a las que llega un tuit de dos cuentas.
Precondiciones	Haber lanzado el tuit.
Postcondiciones	Se visualizan las cuentas a las que ha llegado el tuit.
Flujo normal	1. Iniciar sesión en ambas cuentas. 2. Lanzar tuit. 3. Esperar el tiempo que se desee. 4. Refrescar el Dashboard o esperar al autorefresh.

Tabla 46: CU-03

CU-04	
Nombre	Visualización de comparativa por nombramientos.
Actores	Usuario de la cuenta 1. Usuario de la cuenta 2.
Objetivo	Visualizar el número de nombramientos de dos cuentas.
Precondiciones	Haber lanzado el tuit.
Postcondiciones	Se visualizan el número de nombramientos.
Flujo normal	1. Iniciar sesión en ambas cuentas. 2. Lanzar tuit. 3. Esperar el tiempo que se desee. 4. Refrescar el Dashboard o esperar al autorefresh.

Tabla 47: CU-04

CU-05	
Nombre	Visualización de lugares.
Actores	Usuario.
Objetivo	Visualizar lugares.
Precondiciones	Haber lanzado el tuit.
Postcondiciones	Se podrá visualizar el porcentaje de los lugares desde los que se nombra la marca.
Flujo normal	1. Iniciar sesión en la cuenta que se quiera utilizar. 2. Lanzar tuit. 3. Esperar el tiempo que se desee. 4. Refrescar el Dashboard o esperar al autorefresh.

Tabla 48: CU-05

CU-06	
Nombre	Visualización de cuentas verificadas.
Actores	Usuario.
Objetivo	Visualizar el número de cuentas verificadas desde los que se nombra la marca.
Precondiciones	
Postcondiciones	Se podrá visualizar las cuentas verificadas desde los que se nombra la marca.
Flujo normal	1. Iniciar sesión en la cuenta que se quiera utilizar. 2. Refrescar el Dashboard o esperar al autorefresh.

Tabla 49: CU-06

CU-07	
Nombre	Visualización de top paises.
Actores	Usuario.
Objetivo	Visualizar los países que más han nombrado a la marca.
Precondiciones	
Postcondiciones	Se podrá visualizar los países que más han nombrado la marca.
Flujo normal	1. Iniciar sesión en la cuenta que se quiera utilizar. 2. Refrescar el Dashboard o esperar al autorefresh.

Tabla 50: CU-07

CU-08	
Nombre	Visualización de top palabras.
Actores	Usuario.
Objetivo	Visualizar las palabras más nombradas en relación a la marca.
Precondiciones	
Postcondiciones	Se podrá visualizar las palabras más nombradas en relación a la marca.
Flujo normal	1. Iniciar sesión en la cuenta que se quiera utilizar. 2. Seleccionar la marca con la que queremos la comparativa. 3. Refrescar el Dashboard o esperar al autorefresh.

Tabla 51: CU-08

CU-09	
Nombre	Cambio de cuenta.
Actores	Usuario.
Objetivo	Cambiar la cuenta de Twitter.
Precondiciones	
Postcondiciones	Se cambia la cuenta de Twitter.
Flujo normal	1. Se abre el archivo que contiene la definición del cuerpo de Logstash. 2. Se cambia la cuenta y los parámetros asociados a ella.

Tabla 52: CU-09

CU-10	
Nombre	Cambio de color.
Actores	Usuario.
Objetivo	Cambiar los colores de los gráficos.
Precondiciones	Seleccionar el gráfico que se desea cambiar
Postcondiciones	Cambia el color del gráfico seleccionado.
Flujo normal	1. Se abre el Dashboard asociado como superusuario. 2. Se selecciona el gráfico. 3. Se escoge un nuevo color.

Tabla 53: CU-10

CU-11	
Nombre	Cambio de título de gráficos.
Actores	Usuario.
Objetivo	Cambiar el título de los gráficos.
Precondiciones	Seleccionar el gráfico que se desea cambiar.
Postcondiciones	Cambia el título del gráfico seleccionado.
Flujo normal	1. Se abre el Dashboard asociado como superusuario. 2. Se selecciona el gráfico. 3. Se escoge un nuevo título.

Tabla 54: CU-11

CU-12	
Nombre	Cambio de título de Dashboard.
Actores	Usuario.
Objetivo	Cambiar el título del Dashboard.
Precondiciones	Seleccionar el Dashboard que se desea cambiar.
Postcondiciones	Cambia el título del Dashboard seleccionado.
Flujo normal	1. Se abre el Dashboard asociado como superusuario. 2. Se escoge un nuevo título.

Tabla 55: CU-12

CU-13	
Nombre	Cambio de tipo de gráficos.
Actores	Usuario.
Objetivo	Cambiar el tipo de los gráficos.
Precondiciones	Seleccionar el gráfico que se desea cambiar.
Postcondiciones	Cambia el tipo del gráfico seleccionado.
Flujo normal	1. Se abre el Dashboard asociado como superusuario. 2. Se selecciona el gráfico. 3. Se escoge un nuevo tipo.

Tabla 56: CU-13

4.3 Matriz de trazabilidad.

Para comprobar que existe una relación entre cada requisito con un caso de uso como mínimo, se elabora la siguiente matriz de trazabilidad. En ella se pueden apreciar todas las relaciones obtenidas.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09	CU-10	CU-11	CU-12	CU-13
RF-01	X												
RF-02		X											
RF-03			X										
RF-04				X									
RF-05					X								
RF-06						X							
RF-07							X						
RF-08								X					
RF-09	X												
RF-10		X											
RF-11			X	X									
RF-12			X	X									
RF-13					X								
RF-14						X							
RF-15							X						
RF-16							X						
RF-17								X					
RF-18									X				
RF-19										X			
RF-20											X		
RF-21												X	
RF-22													X

Tabla 57: Matriz RF-CU

5. Implementación e implantación.

En este punto se detallan las características de Implementación del sistema.

5.1 Tecnologías utilizadas.

La solución debe funcionar independientemente del sistema operativo y del explorador.

En este caso en particular se han seleccionado las siguientes herramientas:

- El Sistema operativo sobre el que está instalado *Logstash* es *Debian 9.3.x* La versión de *Logstash*, *Elasticsearch* y *Kibana* es la 5.6 (última versión a 10 de Enero de 2018 estable).
- Para la conexión con la máquina que contiene a *Logstash* se ha utilizado la versión 0.70 de *Putty*.
- Para el almacenamiento de *Elasticsearch+Kibana* se ha utilizado *AWS* plataformado para ELK.
- Para la interfaz de *Kibana* se ha utilizado el navegador *Google Chrome* 64.0.x.

5.2 Metodología a seguir.

Para la creación del *cluster* y posterior indexación se deben seguir los siguientes pasos.

1. Se crea un nuevo *cluster*.

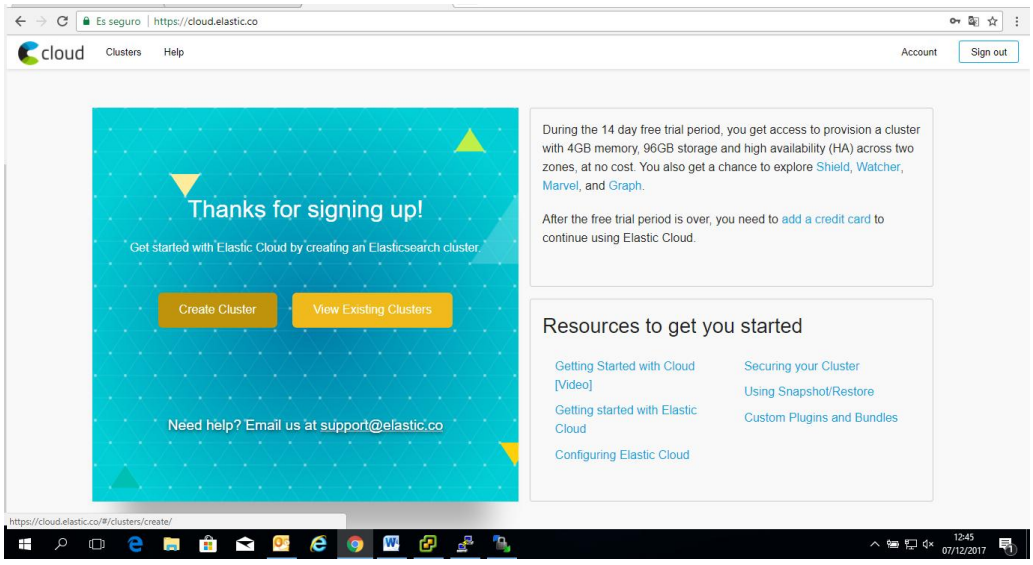

Ilustración 48: Nuevo cluster.

2. Se definen las características del cluster.

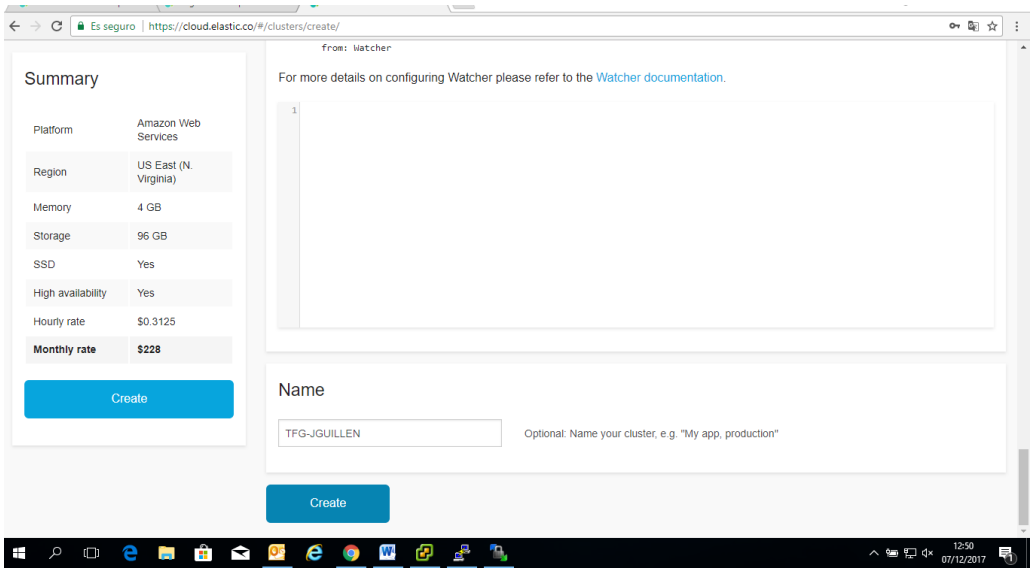

Ilustración 49: Características de cluster.

3. Se copian los datos mostrados para su posterior uso en la conexión con el cluster.

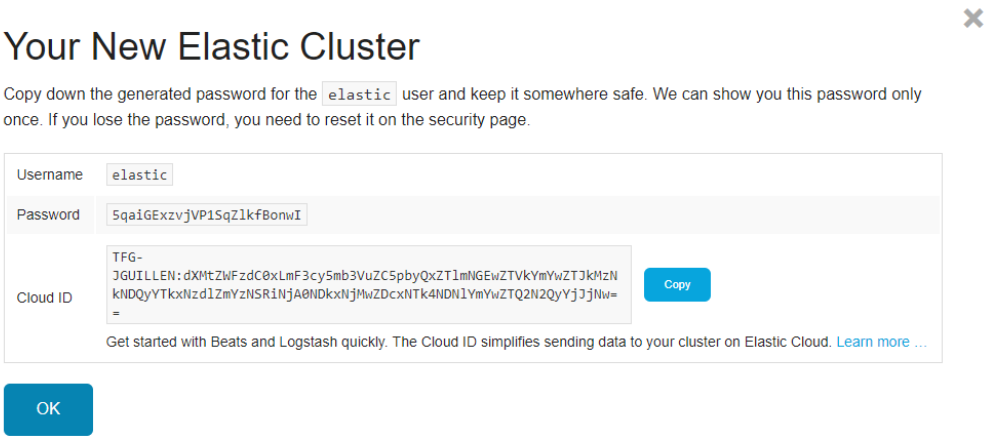

Ilustración 50: Datos cluster.

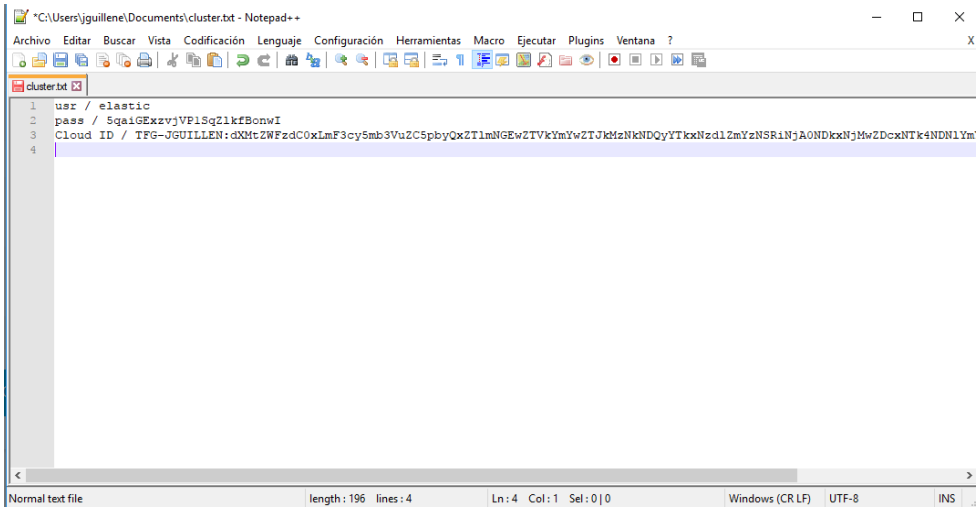

Ilustración 51: Datos guardados.

4. Se comprueba la creación del *cluster*.

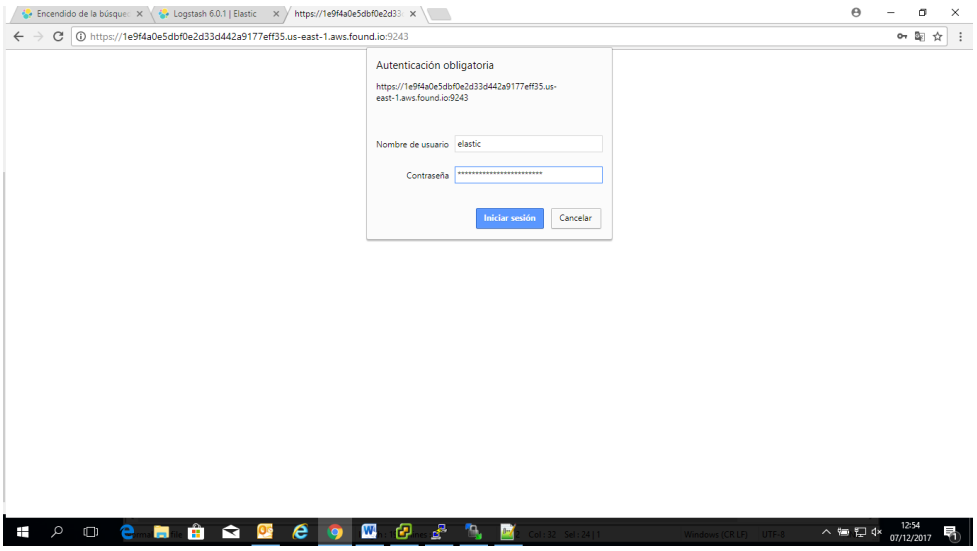

Ilustración 52: Comprobación cluster.

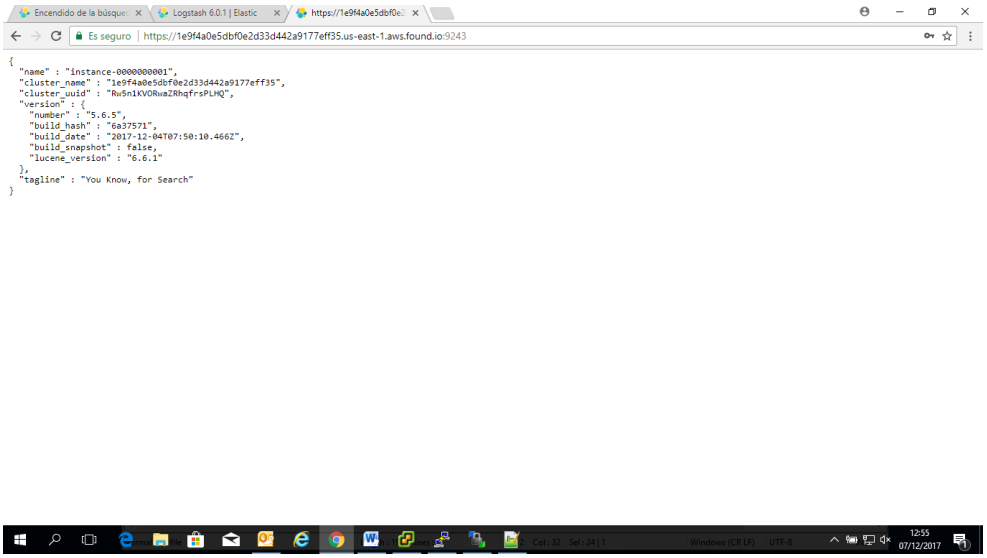

Ilustración 53: Cluster.

5. Se define el índice sobre el que vamos a consultar.

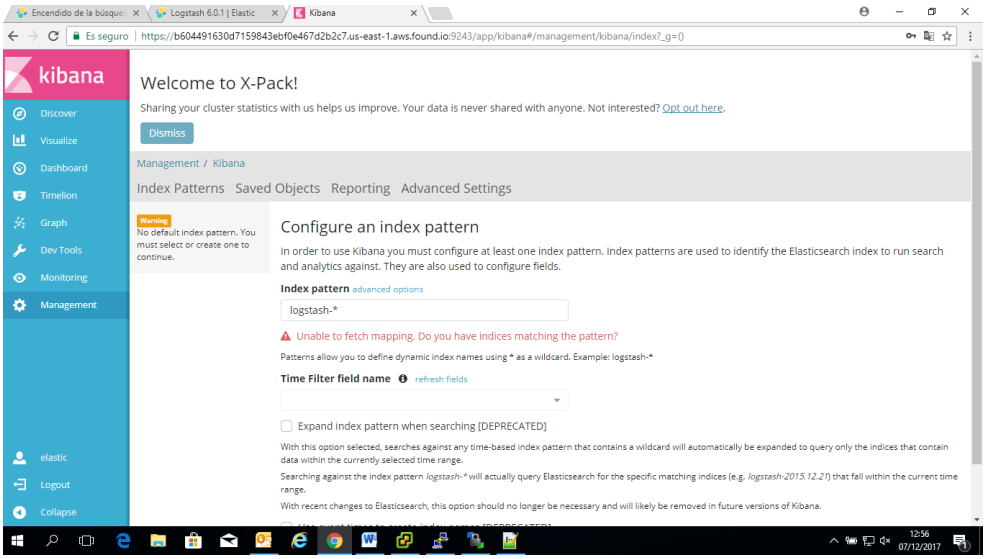

Ilustración 54: Index.

6. Se crea un superusuario para definir todas las características que necesitamos.

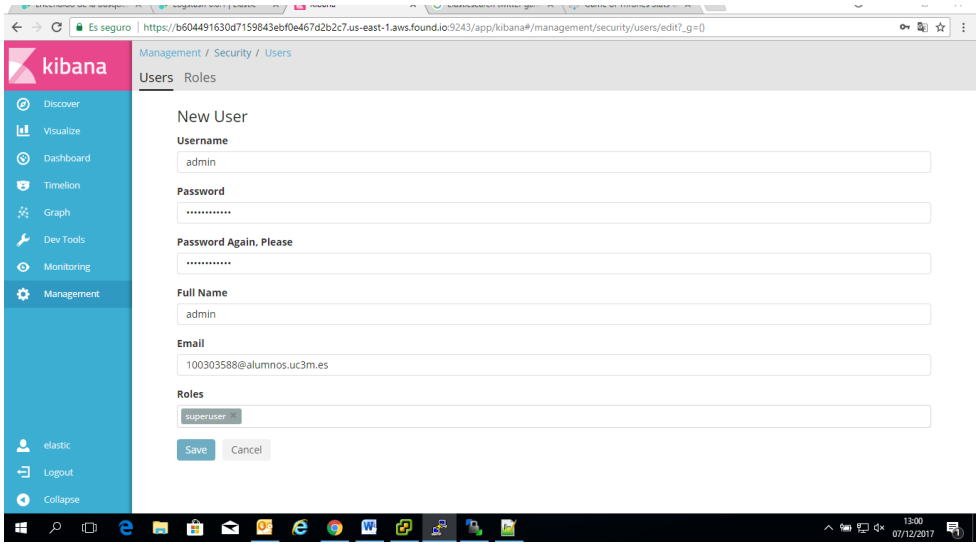

Ilustración 55: Superusuario.

7. Se crea el cuerpo de la extracción de datos de *Logstash.*

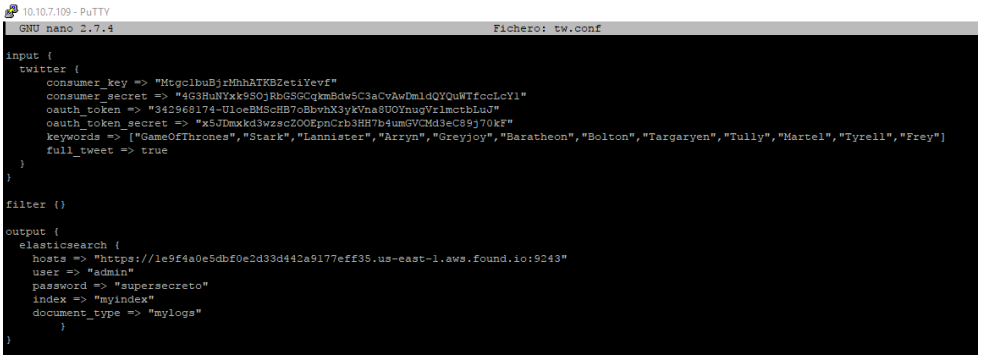

Ilustración 56: Archivo de configuración de Logstash.

8. Se inicia *Logstash.*

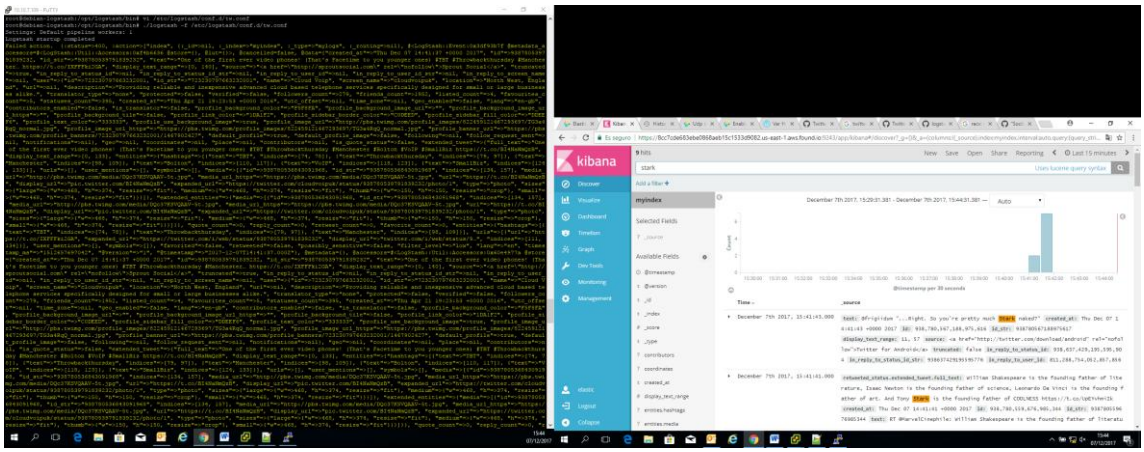

Ilustración 57: Inicio de Logstash.

9. Se crean las visualizaciones requeridas.

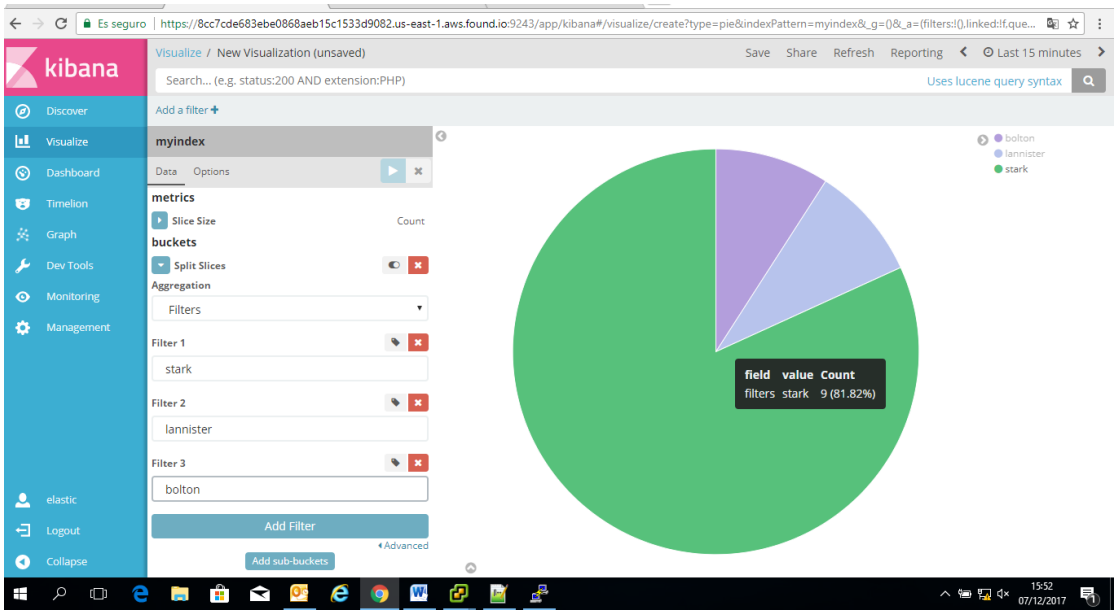

Ilustración 58: Visualización Kibana.

10. Se monta el cuadro de mandos.

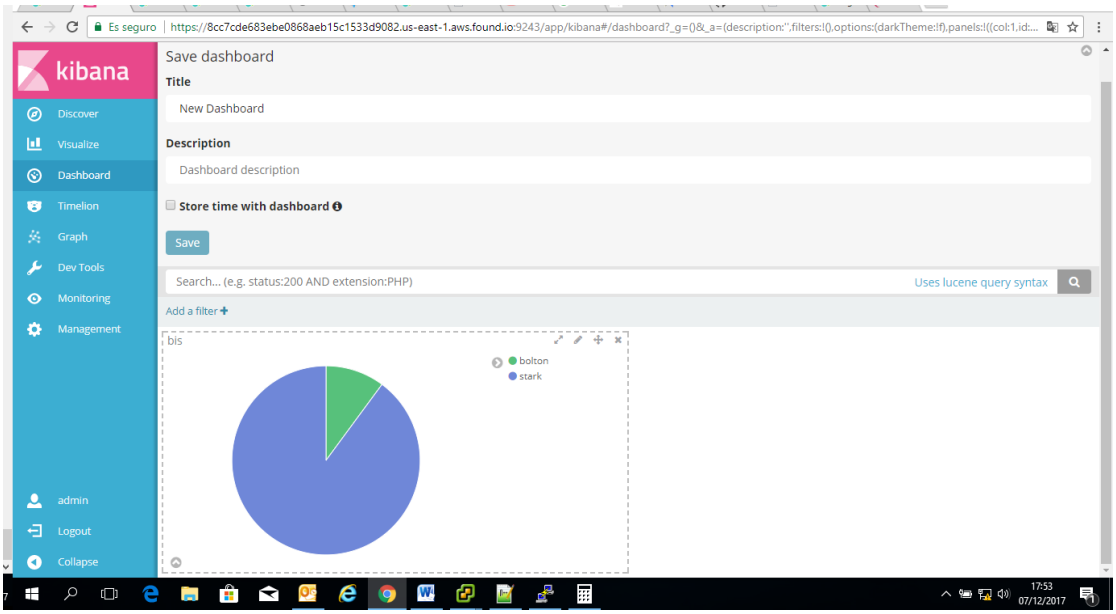

Ilustración 59: Dashboard.

5.2.1 Instalación e inicialización de Logstash.

Para la instalación de *Logstash* se requiere previamente tener *java* en su versión más reciente.

Para instalar *Logstash* bajo *Debian* se siguen los siguientes pasos.

1. Descarga e instalación de clave pública.

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
```

Ilustración 60: Logstash 1.

2. Instalación del paquete *apt-transports-https.*

```
sudo apt-get install apt-transport-https
```

Ilustración 61: Logstash 2.

3. Arrancar la instalación.

```
sudo apt-get update && sudo apt-get install logstash
```

Ilustración 62: Logstash 3.

Una vez instalado se debe comprobar que todo ha salido correctamente, esto se puede realizar mediante una definición simple del siguiente esquema:

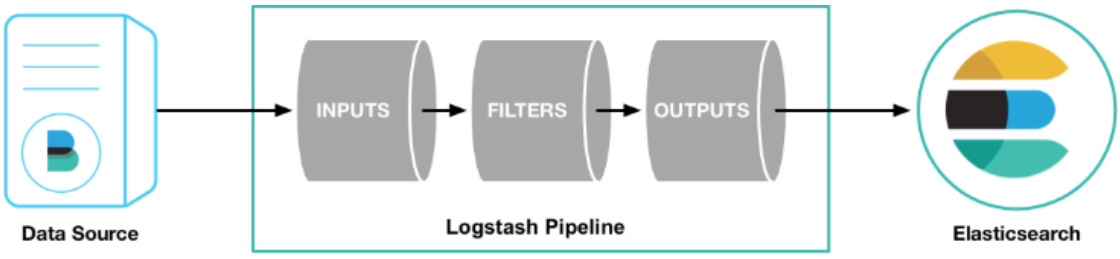

Ilustración 63: Esquema Logstash.

La definición más simple es la de la ilustración siguiente.

```
cd logstash-5.6.7
bin/logstash -e 'input { stdin { } } output { stdout {} }'
```

Ilustración 64: Logstash 4.

La tubería de ejemplo toma la entrada estándar *stdin* y lo mueve a la salida estándar *stdout* de forma estructurada, agregando una marca de tiempo y la *dirección IP*.

```
hello world
2013-11-21T01:22:14.405+0000 0.0.0.0 hello world
```

Ilustración 65: Logstash 5.

Para este proyecto la definición será la mostrada a continuación.

```
input {
  twitter {
      consumer_key => "Mtgc1buBjrMhhATKBZetiYevf"
      consumer_secret => "4G3HuNYxk9SOjRbGSGCqkmBdw5C3aCvAwDmldQYQuWTfccLcY1"
      oauth_token => "342968174-UloeBMScHB7oBbvhX3ykVna8UOYnugVrlmctbLuJ"
      oauth_token_secret => "x5JDmxkd3wzscZOOEpnCrb3HH7b4umGVCMd3eC89j70kF"
      keywords => ["marca"]
      full_tweet => true
  }
}

filter {}

output {
  elasticsearch {
    hosts => "https://ac532bd3aa91dc207b2c1b90da121527.us-east-1.aws.found.io:9243"
    user => "admin"
    password => "pass"
    index => "myindex"
    document_type => "mylogs"
      }
}
```

Ilustración 66: Cuerpo Logstash.

5.2.2 Instalación e inicialización de Elasticsearch y Kibana.

Para el montaje en la nube de estas dos herramientas se tienen que seguir los siguientes pasos:

1. Registro en *Elastic cloud* y creación del *cluster*.

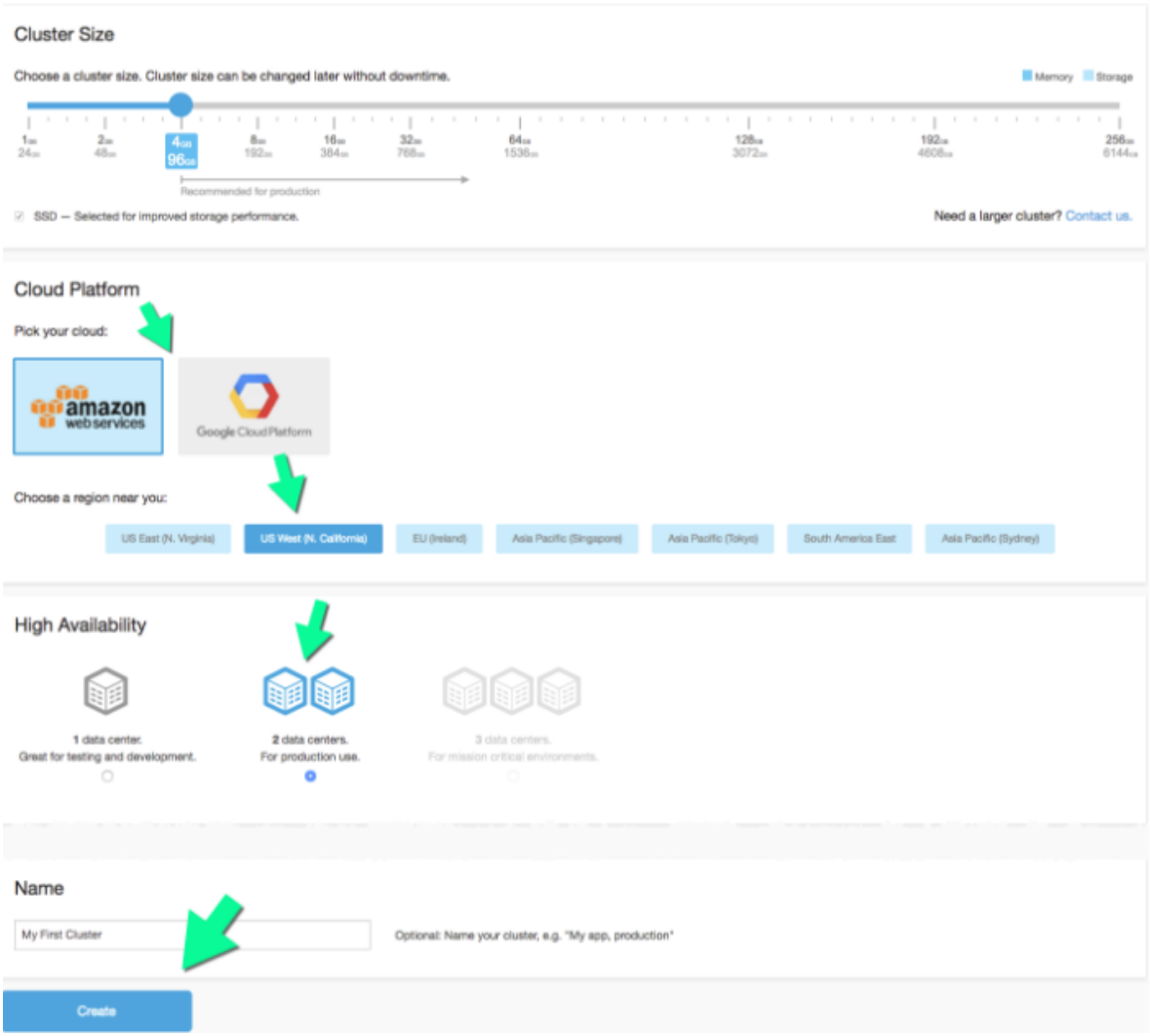

Ilustración 67: Registro Elastic Cloud.

2. Conexión al *cluster* para comprobar su correcto funcionamiento.


```
{
  "name" : "instance-0000000002",
  "cluster_name" : "16bee70bb9c9f6cd673a0567bbe0cc19",
  "cluster_uuid" : "LiiTMMEmSoOanddQZ79Kiw",
  "version" : {
    "number" : "6.2.1",
    "build_hash" : "7299dc3",
    "build_date" : "2018-01-02T19:34:26.990113Z",
```

```
    "build_snapshot" : false,
    "lucene_version" : "7.2.1",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```


5.3 Interfaz de Kibana.

En la parte izquierda de *Kibana* se pueden observar varios apartados que se deben analizar.

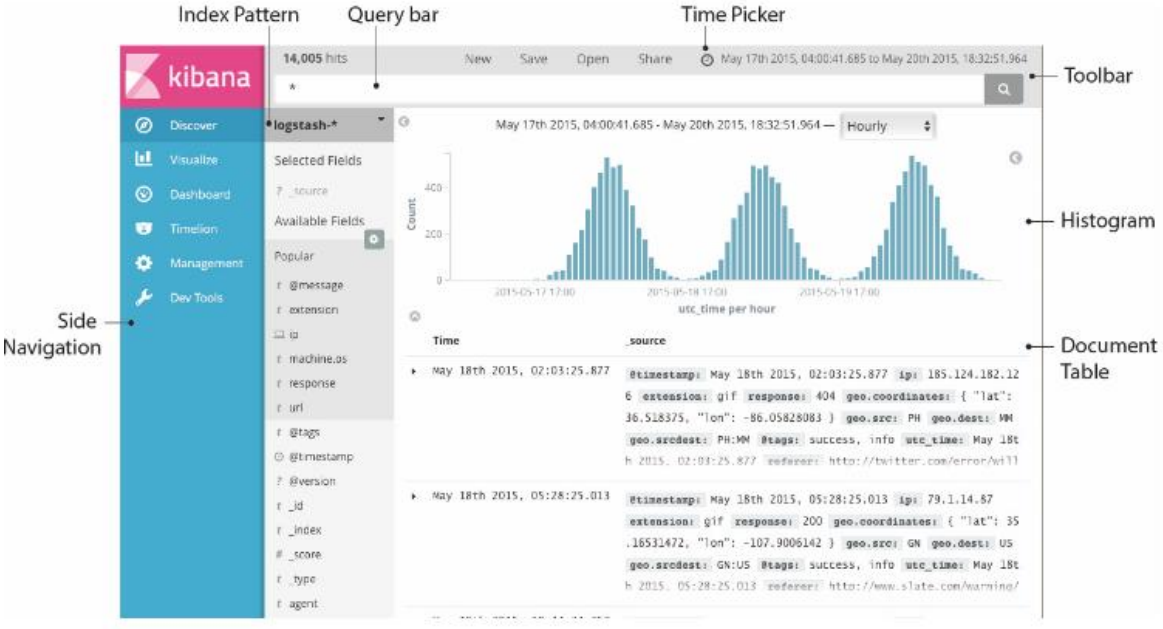

Ilustración 68: Interfaz Kibana.

1. **Discover.**

   Se puede explorar los datos de forma interactiva, tanto a nivel de documento como a nivel de índice. Se pueden realizar consultas directamente, filtrar resultados y ver los datos seleccionados.

2. **Visualize.**

   Permite crear visualizaciones de los datos en el índice de *Elasticsearch*. Las visualizaciones de *Kibana* están basadas en consultas de *Elasticsearch*. Al modificar las métricas obtenidas de diferentes formas para extraer la información necesaria se pueden crear gráficos que muestren tendencias y saltos que se deben conocer.

3. **Dashboard.**

   Es una colección de visualizaciones guardadas. Incluye un panel propio de edición.

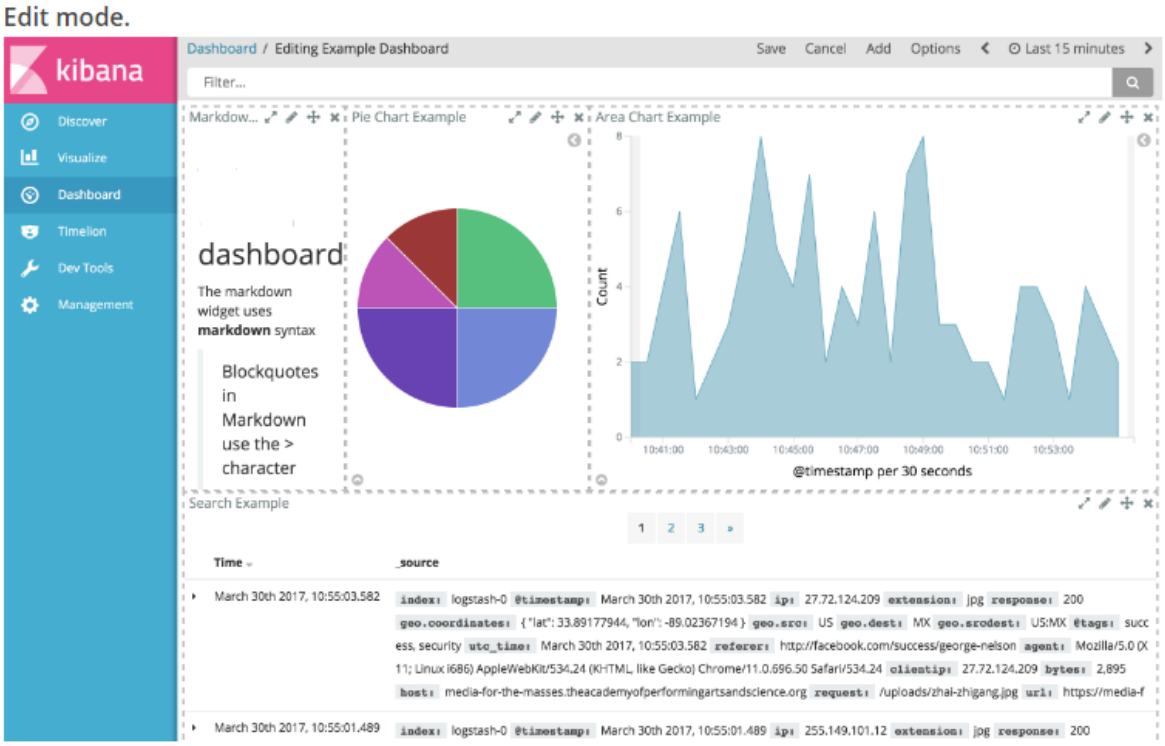

Ilustración 69: Dashboard

4. **Timelion.**

   Enfocado en series temporales.

5. **Dev Tools.**

   Contiene herramientas de desarrollo para interactuar con los datos.

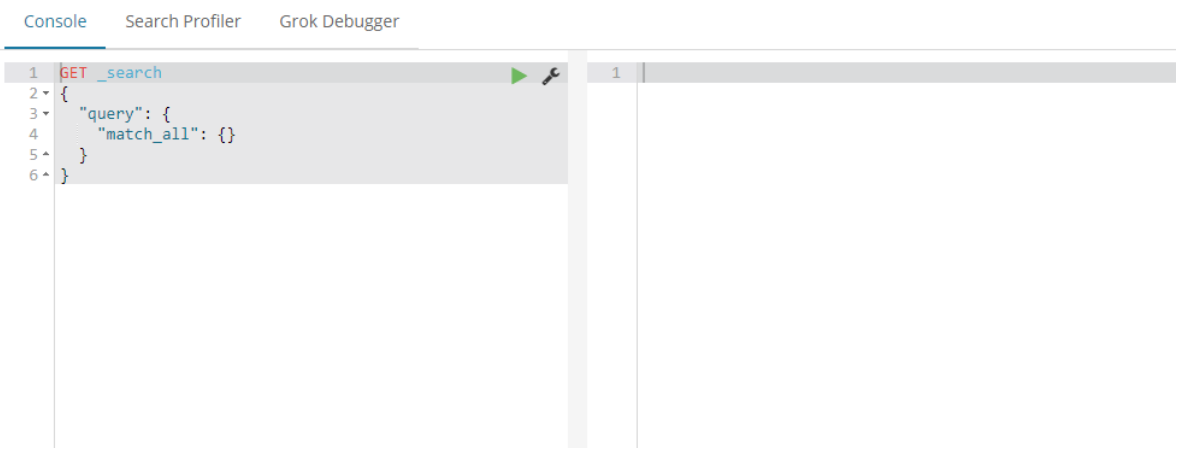

Ilustración 70: Dev tools.

5.4 Pruebas.

En este punto se detallarán las diferentes pruebas realizadas al sistema para comprobar que funciona correctamente siguiendo la tabla que se muestra a continuación.

Código de prueba PRU-XX.	**Prueba.**	**Descripción.**

Ilustración 71: Pruebas.

Los atributos que definen a la prueba son:

- Código de prueba: Código unívoco que define una prueba.
- Prueba: Nombre de la prueba.
- Descripción.

Todas las pruebas se deben realizar bajo cuentas con 0 seguidores y ningún tipo de impacto, con palabras aleatorias para poder comprobar el correcto funcionamiento, teniendo una cuenta para lanzar tuits y otra para visualizarlos.

Código de prueba.	Prueba.	Descripción.
PRU-01	Visualización de cuentas.	El usuario al lanzar un tuit y refrescar *Kibana* es capaz de ver las cuentas a las que llega el tuit.
PRU-02	Visualización de nombramientos	El usuario debe marcar un tiempo de inicio y un tiempo final. Tras refrescar *Kibana* podrá ver el número de veces que ha sido nombrada.
PRU-03	Visualización de comparativa por cuentas.	El usuario al lanzar un tuit y refrescar *Kibana* es capaz de ver la comparativa de cuentas a las que llega respecto a otro tuit.
PRU-04	Visualización de comparativa por nombramientos.	El usuario al refrescar *Kibana* es capaz de ver la comparativa de nombramientos de su cuenta frente a otra.
PRU-05	Visualización de lugares.	El usuario al lanzar un tuit y refrescar *Kibana* es capaz de ver el lugar desde el que se visualiza el tuit.
PRU-06	Visualización de cuentas verificadas.	El usuario al refrescar *Kibana* es capaz de ver el número de seguidores cuyas cuentas están verificadas.
PRU-07	Visualización de top países.	El usuario al refrescar *Kibana* es capaz de ver los cinco países que más han nombrado la marca.

PRU-08	Visualización de top palabras.	El usuario al refrescar *Kibana* es capaz de ver las palabras que más se nombran junto con la marca.
PRU-09	Cambio de cuenta.	El usuario es capaz de cambiar de cuenta.
PRU-10	Cambio de color de gráficos.	El usuario al cambiar el color de un gráfico y refrescar la página ve efectivo el cambio.
PRU-11	Cambio de título de gráficos.	El usuario al cambiar el título de los gráficos y refrescar la página ve efectivo el cambio.
PRU-12	Cambio de título de *dashboard*.	El usuario al cambiar el título del *dashboard* y refrescar la página ve efectivo el cambio.
PRU-13	Cambio de tipo de gráfico.	El usuario al cambiar el tipo de los gráficos y refrescar la página ve efectivo el cambio.

Tabla 58: Descripción de pruebas.

A continuación se muestra la matriz de relación entre los casos de uso y las pruebas realizadas.

	PRU-01	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**	**PRU-01**
CU-01	X												
CU-01		X											
CU-01			X										
CU-01				X									
CU-01					X								
CU-01						X							
CU-01							X						
CU-01								X					
CU-01									X				
CU-01										X			
CU-01											X		
CU-01												X	
CU-01													X

Tabla 59: Matriz PRU-CU.

6.0 Planificación y presupuesto.

En este punto se detallará la planificación desarrollada durante el proyecto haciendo uso de dos diagramas de Gantt, uno con la planificación y otro con el tiempo real.

Además se establece un presupuesto para el proyecto de forma real, sin periodos de prueba y descartando el ámbito académico.

6.1 Modelo del ciclo de vida.

En este punto se observa el modelo elegido, en este caso un modelo en cascada con retroalimentación en el que en cada uno de los pasos se puede avanzar o retroceder al punto de partida.

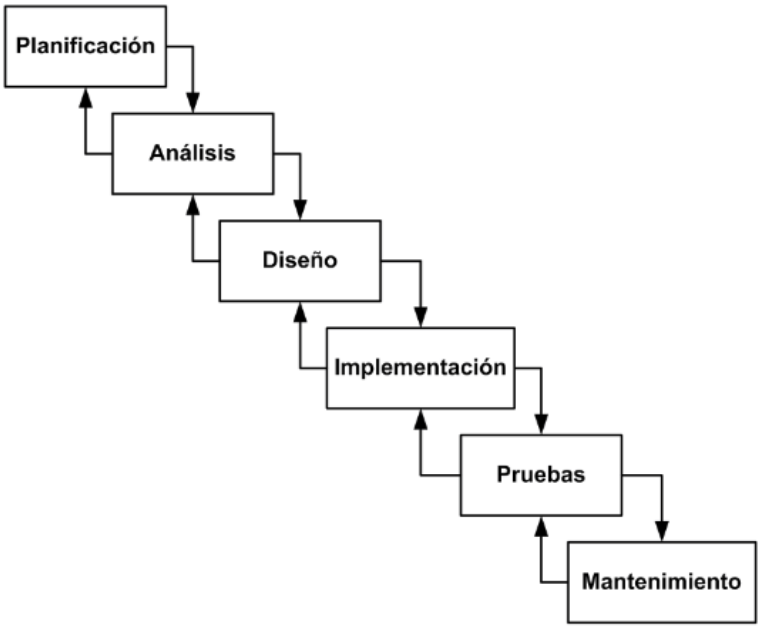

Ilustración 72: Planificación en cascada.

Esto permite que tras encontrar fallos en cada una de las etapas se pueda retroceder de forma escalable para realizar cambios en las etapas anteriores.

6.2 Planificación del proyecto.

Este trabajo se clasifica dentro de 3 grandes bloques, más la escritura de la memoria. Estos son: estudio previo o análisis, diseño de la solución e implementación. Los subapartados se detallarán a continuación.

- Análisis.
    - Estudio de las diferentes formas de abordar el problema de la ingesta masiva de datos en Twitter.

        - Estudio de la solución sin herramienta.
        - Estudio de la solución con herramientas de BI.
    - Estudio del impacto socio-económico.
    - Estudio de las diferentes herramientas de BI.
        - Estudio del *stack ELK*.
        - Estudio de otras alternativas.
    - Periodo de aprendizaje del *stack ELK*.
- Diseño de la solución.
    - Redacción de los requisitos.
    - Diseño del esquema de atributos de la *API* de *Twitter*.
    - Diseño de las visualizaciones.
    - Diseño de *dashboard*.
- Implementación.
    - Definición del sistema de implantación.
    - Instalación de *Logstash*.
    - Implementación del archivo de configuración de *Logstash*.
    - Implementación de *Elasticsearch* en el sistema *Elastic Cloud*.
    - Implementación de *Kibana* en el sistema *Elastic Cloud*.
- Redacción de la memoria.

6.2.1 Planificación inicial.

El plazo inicial para la realización de este proyectó se estimó el 1 de Julio de 2017 para estar finalizado en un máximo de 8 meses, fecha de presentación del proyecto.

Para ello se llevó a cabo un sistema que dividía los procesos en estimación de aprendizaje y/o dificultad. A continuación se muestra una tabla con estas características cuyos datos numéricos se encuentran en un rango [1,4].

Fases del proyecto.	Dificultad.	Dificultad de aprendizaje.
Análisis.		
Estudio de la solución sin herramienta.	2	-
Estudio de la solución con herramientas de BI.	4	-
Estudio del impacto socio-económico.	1	-
Estudio del *stack ELK.*	4	-
Estudio de otras alternativas.	2	-
Diseño de la solución.		
Redacción de los requisitos.	2	-
Diseño del esquema de atributos de la API de Twitter.	3	-
Diseño de las visualizaciones.	3	-
Diseño de *dashboard.*	2	-

Implementación.		
Definición del sistema de implantación.	2	-
Instalación de *Logstash*.	4	4
Implementación del archivo de configuración de *Logstash*.	1	4
Implementación de *Elasticsearch* en el sistema *Elastic Cloud*.	1	4
Implementación de *Kibana* en el sistema *Elastic Cloud*.	1	4
Memoria.		
Redacción de la memoria.	1	-

Tabla 60: Dificultad.

Se puso al seguir este sistema 4 días por unidad de dificultad excepto en la memoria, cuya redacción es continua en el tiempo desde el inicio del proyecto hasta el final.

Por lo tanto la planificación inicial fue la que se expone a continuación.

Tarea.	Fecha de inicio.	Fecha fin.	Duración.
Análisis.			
Estudio de la solución sin herramienta.	01/07/2017	09/07/2017	8 días.
Estudio de la solución con herramientas de BI.	10/07/2017	26/07/2017	16 días.
Estudio del impacto socio-económico.	27/07/2017	31/07/2017	4 días.
Estudio del *stack ELK.*	1/08/2017	17/08/2017	16 días.
Estudio de otras alternativas.	18/08/2017	26/08/2017	8 días.
Diseño de la solución.			
Redacción de los requisitos.	27/08/2017	4/09/2017	8 días.
Diseño del esquema de atributos de la *API* de *Twitter.*	5/09/2017	17/09/2017	12 días.

Diseño de las visualizaciones.	18/09/2017	30/09/2017	12 días.
Diseño de *dashboard*.	1/10/2017	9/10/2017	8 días.
Implementación.			
Definición del sistema de implantación.	10/10/2017	18/10/2017	8 días.
Instalación de *Logstash*.	19/10/2017	20/11/2017	32 días.
Implementación del archivo de configuración de *Logstash*.	21/11/2017	10/12/2017	20 días.
Implementación de *Elasticsearch* en el sistema *Elastic Cloud*.	11/12/2017	31/12/2017	20 días.
Implementación de *Kibana* en el sistema *Elastic Cloud*.	1/01/2018	20/01/2018	20 días.

Memoria.			
Redacción de la memoria.	1/07/2017	20/01/2018	6 meses y 20 días.

Tabla 61: Planificación inicial.

A continuación se muestra el *Gannt* dividido de la planificación inicial.

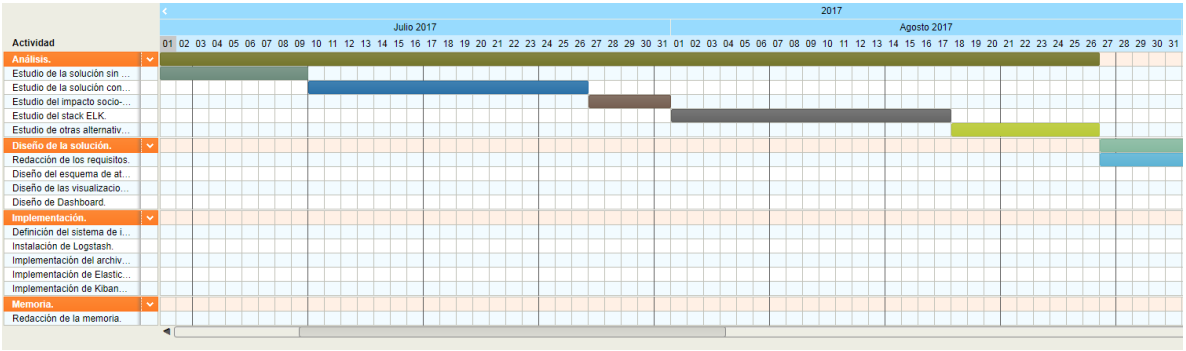

Ilustración 73: Gantt inicial 1.

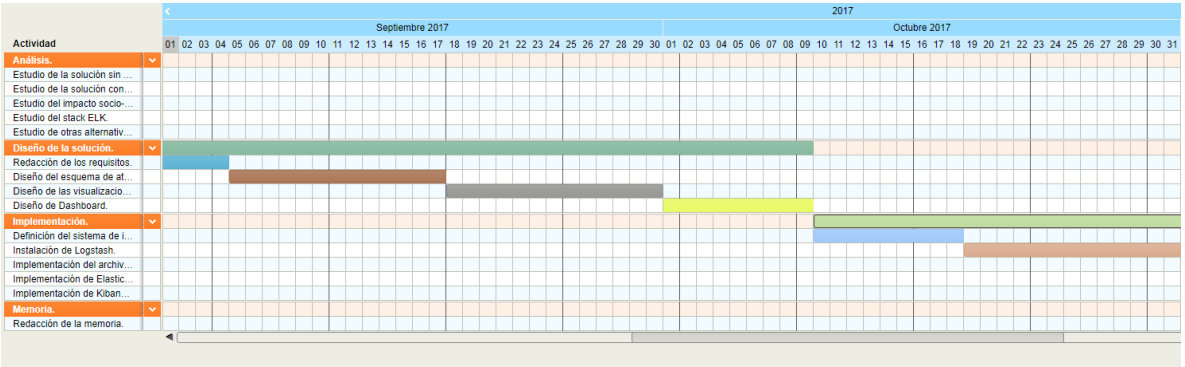

Ilustración 74: Gantt inicial 2.

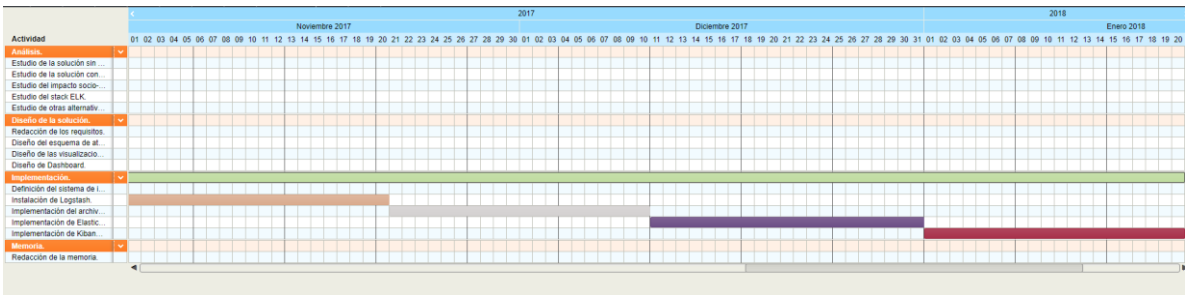

Ilustración 75: Gantt Inicial 3.

6.2.1 Planificación final.

Tras el desarrollo del proyecto el tiempo real difiere en algunos puntos, por lo que se reestructura tanto la tabla de tiempos como el gráfico de *Gantt*, dando el siguiente resultado.

Tarea.	**Fecha de inicio.**	**Fecha fin.**	**Duración.**
Análisis.			
Estudio de la solución sin herramienta.	01/07/2017	09/07/2017	8 días.
Estudio de la solución con herramientas de BI.	10/07/2017	22/07/2017	12 días.
Estudio del impacto socio-económico.	23/07/2017	30/07/2017	7 días.
Estudio del *stack ELK.*	31/07/2017	19/08/2017	19 días.
Estudio de otras alternativas.	19/08/2017	23/08/2017	4 días.
Diseño de la solución.			
Redacción de los requisitos.	24/08/2017	4/09/2017	11 días.

Diseño del esquema de atributos de la API de Twitter.	5/09/2017	12/09/2017	7 días.
Diseño de las visualizaciones.	13/09/2017	22/09/2017	9 días.
Diseño de *dashboard*.	23/09/2017	3/10/2017	10 días.
Implementación.			
Definición del sistema de implantación.	4/10/2017	10/10/2017	6 días.
Instalación de *Logstash*.	11/10/2017	5/11/2017	25 días.
Implementación del archivo de configuración de *Logstash*.	6/11/2017	25/11/2017	19 días.
Implementación de *Elasticsearch* en el sistema *Elastic Cloud*.	26/11/2017	15/12/2017	19 días.

Implementación de *Kibana* en el sistema *Elastic Cloud.*	16/12/2017	3/01/2018	18 días.
Memoria.			
Redacción de la memoria.	1/07/2017	3/01/2018	6 meses y 3 días.

Ilustración 76: Planificación real.

Se puede observar que la estimación inicial fue al alza ya que se tardaron 15 días menos en realizar el proyecto de lo estimado. A continuación se muestra el gráfico Gantt del tiempo real.

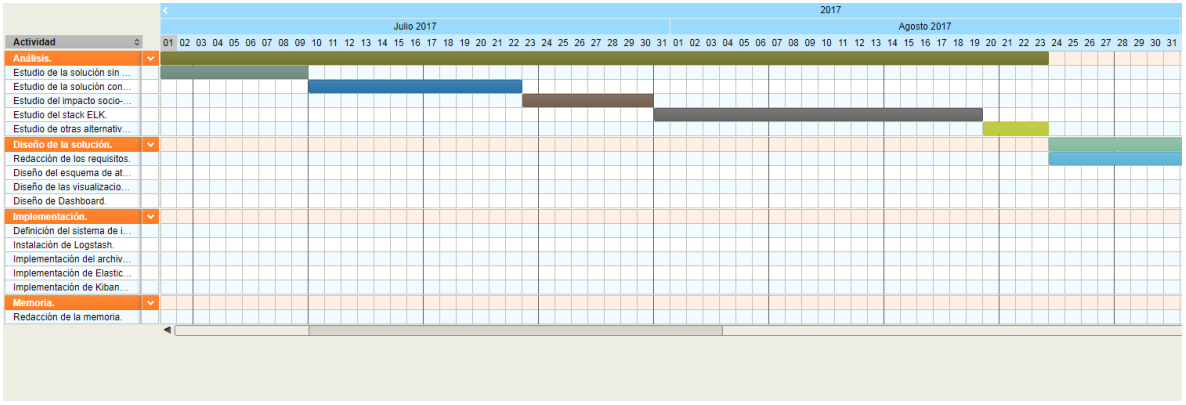

Ilustración 77: Gantt real 1.

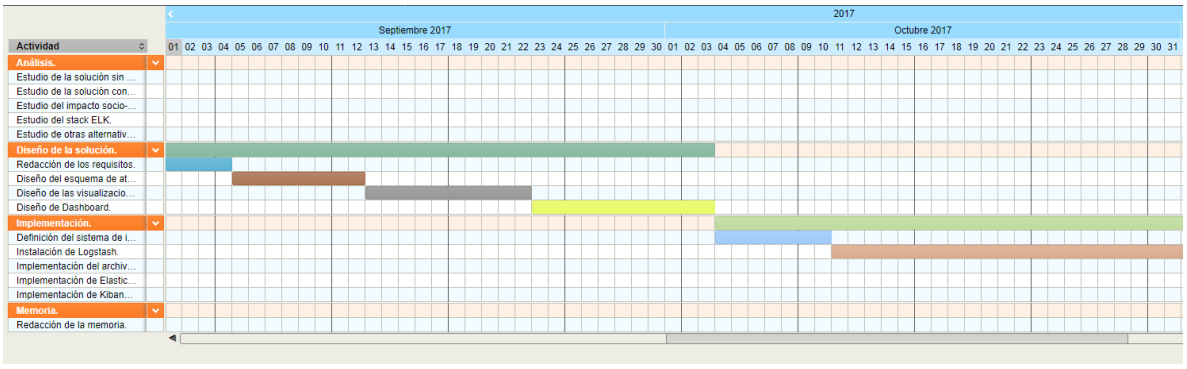

Ilustración 78: Gantt real 2.

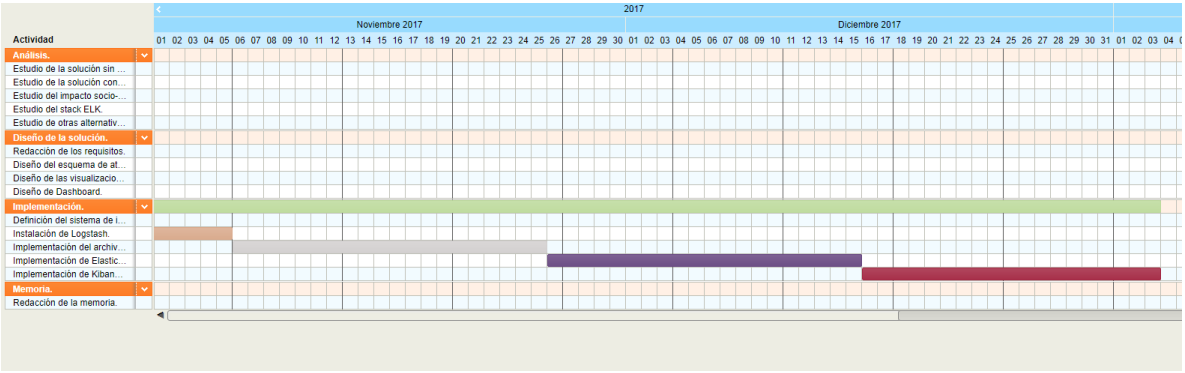

Ilustración 79: Gantt real 3.

6.3 Presupuesto.

En este apartado se define el presupuesto y el coste del desarrollo de este proyecto en un ámbito real de implementación, con un mantenimiento de 5 años.

6.3.1 Costes de personal.

Este proyecto necesita bien de una persona con conocimientos de programación y de marketing o bien dos personas con conocimientos separados. Lo más común es encontrar expertos de cada tema, por lo que se tendrá por un lado el coste del programador para la implementación y el mantenimiento y por otro lado el experto en marketing que se encargará de la revisión y extracción del conocimiento.

Según narra [10] la media salarial para un programador que conoce el *stack ELK* y no necesita formación es de 27.000 € brutos/año, mientras que la media para un subdirector de marketing encargado de la toma de decisiones y de la visualización del *dashboard* es de 50.000€ al año.

A estos costes habrá que añadir los costes de SS que se muestran en la siguiente tabla.

- Tarea de implantación del sistema.

Programador		**2.250€/mes**
Seguridad social	Porcentaje	Coste
Total	35%	787,5€/mes
Total		
3.037,5€/mes		

Tabla 62: Coste programador.

Según el proyecto, la implantación se estima en 88 días, por lo que se tiene contratado al programador 3 meses, sumando un total de 9112,5 €.

El mantenimiento se estima en un 10% del salario del programador, por lo que para los 5 años se requiere 303,75 €/mes * 57 meses de mantenimiento, lo que equivale a 17.313,75 €.

En total, el presupuesto para la tarea del programador se cifrará en 26.426,25€.

- Tarea de marketing.

Subdirector de marketing y publicidad.		**4.166€/mes**
Seguridad social.	Porcentaje.	Coste.
Total.	35%	1.458,1€/mes
Total		
5.624,1€/mes		

Tabla 63: Coste marketing.

Para los 5 años de la tarea del subdirector de publicidad se requiere 5.624,1€/mes * 59 meses, lo que equivale a 331821,9€.

En total el coste de personal es de 358248,15€ para la implantación y 5 años de mantenimiento.

6.3.2 Coste de material y licencias.

El material necesario para la realización de un proyecto es una máquina física donde instalar *Debian 9.3* junto a *Logstash*. Esto para una máquina estándar se estima en 500€.

En cuanto a licencias todo el software utilizado es gratuito.

6.3.3 Coste de luz y ADSL.

Los costes de luz y ADSL se detallan a continuación.

Tipo.	**Precio Mensual.**
Fibra Óptica 100 Mb	50€/mes
Total 5 años.	
2.950€ (I.V.A. incl.)	

Tabla 64: Coste ADSL.

Tipo	Potencia media consumida.	Tiempo de utilización.	Precio medio KWh.	Coste mensual.
Consumo de la máquina física.	180W	24h/día	0,146€/KWh	18€
Total 5 años.				
708€ (I.V.A. incl.)				

Tabla 65: Coste luz.

Por lo que el coste total de luz + *ADSL* se estima en 3.658€ (I.V.A. incl.).

6.3.4 Coste del sistema Elastic Cloud bajo AWS.

El coste de utilización del sistema en la nube es de $1748/mes (ver punto 4.1.3), lo que equivale a 1409 € actualmente.

El total para el mantenimiento de los 5 años es 83.131€ (I.V.A. incl.).

6.3.5 Coste total.

El coste total del proyecto queda reflejado en la siguiente tabla.

Tipo.	Coste.
Personal.	358.248,15€
Material.	500€
Luz + ADSL.	3.658€
Elastic Cloud.	83.131€
Total.	
445.537,15 (I.V.A. incl.).	

Tabla 66: Coste total.

El coste total del proyecto se estima en **445.537,15** (I.V.A. incl.).

7.0 Conclusiones y líneas futuras.

En este apartado se concluye el trabajo aportando las conclusiones finales del mismo, así como líneas de trabajo futuro y líneas alternativas.

7.1 Conclusiones.

7.1.1 Sistema ELK.

El trabajo gira entorno a Elasticsearch y la pila de herramientas que lo engloban, por lo que se puede observar una tendencia a utilizar motores de búsqueda de texto en sistemas no relacionales, cuya ingesta de datos es bastante grande.

El sistema *ELK* no es sencillo de entender en un inicio, ya que cada herramienta puede ser utilizada de forma unilateral, a la vez que son parte de un todo. Debido a esto puede resultar dificultoso entender cómo realizar el trabajo de encaje de los datos.

No obstante, cuando se entiende el proceso y se observan los *logs*, investigando los errores se puede llegar a incluso subsanar y ser parte de la comunidad que crea el sistema, aportando soluciones nuevas.

Se ha escogido *ELK* por ser sin duda el sistema que más se acerca al planteamiento inicial, en el cual se necesitaba extraer los datos de *Twitter* y transformarlos (*Logstash*), posteriormente almacenarlos de forma estructurada (*Elasticsearch*) y finalmente mostrar un *dashboard* (*Kibana*).

7.1.2 Instalación ELK.

Al ser un *stack* relativamente nuevo con millones de usuarios se encuentra en constante cambio. Esto requiere una actualización de conocimiento constantes por lo que los manuales de instalación están totalmente obsoletos.

Para la instalación de *ELK* en un inicio se procuró utilizar la versión más reciente, no obstante se tuvo que montar sobre la versión 5.6, al ser la versión estable más actualizada.

La máquina física donde se ha encontrado *Logstash* durante todo el proceso ha sido un servidor, por lo que además han hecho falta herramientas como *Putty* y *VMware* con su correspondiente aprendizaje.

7.1.2 Proceso de desarrollo.

En un principio se pensó en desarrollar la idea para Logs generados por ordenadores de la universidad Carlos III de Madrid. Tras este planteamiento inicial se pasó al proceso de análisis viendo que se habían escrito múltiples artículos sobre *ELK* y otras herramientas de BI para el procesamiento de logs, por lo que este trabajo era

insuficiente. Entonces nació la idea de la necesidad de una persona en compararse en *RRSS* con los demás, en cuanto a números de seguidores, reacciones antes las publicaciones, etc.

Todo ello finalmente evolucionó hasta el punto de reflexión sobre quiénes son los más interesados en saber cierto tipo de información con cantidades masivas de datos, por lo que se decantó por utilizarlo en sistemas de publicidad en *Twitter*, es decir, *tuits* sobre una marca en concreto.

Tras este proceso se inició la puesta en marcha, creando una cuenta de *Twitter* y una aplicación *Twitter* para recoger los *Tokens* necesarios.

Una vez establecida la conexión con Twitter y enlazado con *Elasticsearch*, es cuestión de práctica y lectura de manual poder crear cualquier tipo de visualización que desee con los datos obtenidos.

7.1.3 Conclusiones personales.

Personalmente este proyecto ha sido el más grande en el que me he visto involucrado, haciendo que pudiera vivir de primera mano desde el minuto cero la evolución de un proyecto, aprendiendo desde la instalación de un software totalmente desconocido para mí hasta poder aportar soluciones a la comunidad de *ELK* actualmente.

Volvería sin duda a elegir este proyecto, quizá enfocándolo desde otros puntos de vista, pero siempre bajo *ELK* y el mismo apoyo de compañeros y tutor.

7.2 Trabajos futuros

Hay múltiples opciones en cuanto a trabajos futuros, con la misma idea del proyecto se pueden cambiar los requisitos, haciendo importantes otras ideas aquí no obtenidas, por ejemplo la hora en la que más se nombra a una cuenta.

Además gracias a que no es un sistema único se pueden combinar con otros como *Beats* o *Grafana.*

En este aspecto *Beats* podría proporcionar los *tuits* a *Elasticsearch* y *Grafana* pintar bajo este esquema.

Un trabajo futuro que no tenía cabida por la dificultad y tiempo del proyecto es la incorporación de análisis de sentimientos de los tuits, puesto que existen herramientas que se encargan de dado un texto analizar el sentimiento que describe. Por lo tanto, se podría utilizar *Logstash* para extraer los tuits, proporcionárselo a una de estas herramientas como plugin en la definición del filtro de su cuerpo y dar un nuevo parámetro en función del estado de ánimo.

FIN DEL DOCUMENTO.

Bibliografía.

[1] (Scientia et Technica Año XVI, No 44, Abril de 2010. Universidad Tecnológica de Pereira. ISSN 0122-1701)

[2] Inteligencia de negocios y estado del arte: https://www.researchgate.net/publication/277231717_Inteligencia_de_negocios_Estado_del_arte

[3] (Zúmel 2008)

[4] https://tristanelosegui.com/2014/10/27/que-es-y-para-que-sirve-un-dashboard/

[5] https://www.brandwatch.com/es/2016/06/44-estadisticas-twitter-2016/

[6] Psicología del color para la elección de colores en los gráficos del Dashboard: http://www.eartvic.net/~mbaurierc/materials/20%20Selectivitat/Psicologia%20del%20color.pdf

[7] https://logz.io/blog/fluentd-logstash/

[8] https://logz.io/blog/solr-vs-elasticsearch/

[9] https://www.elastic.co